1995年3月1日発行（毎月1回1日発行）第14巻3号通巻155号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

Oh!X

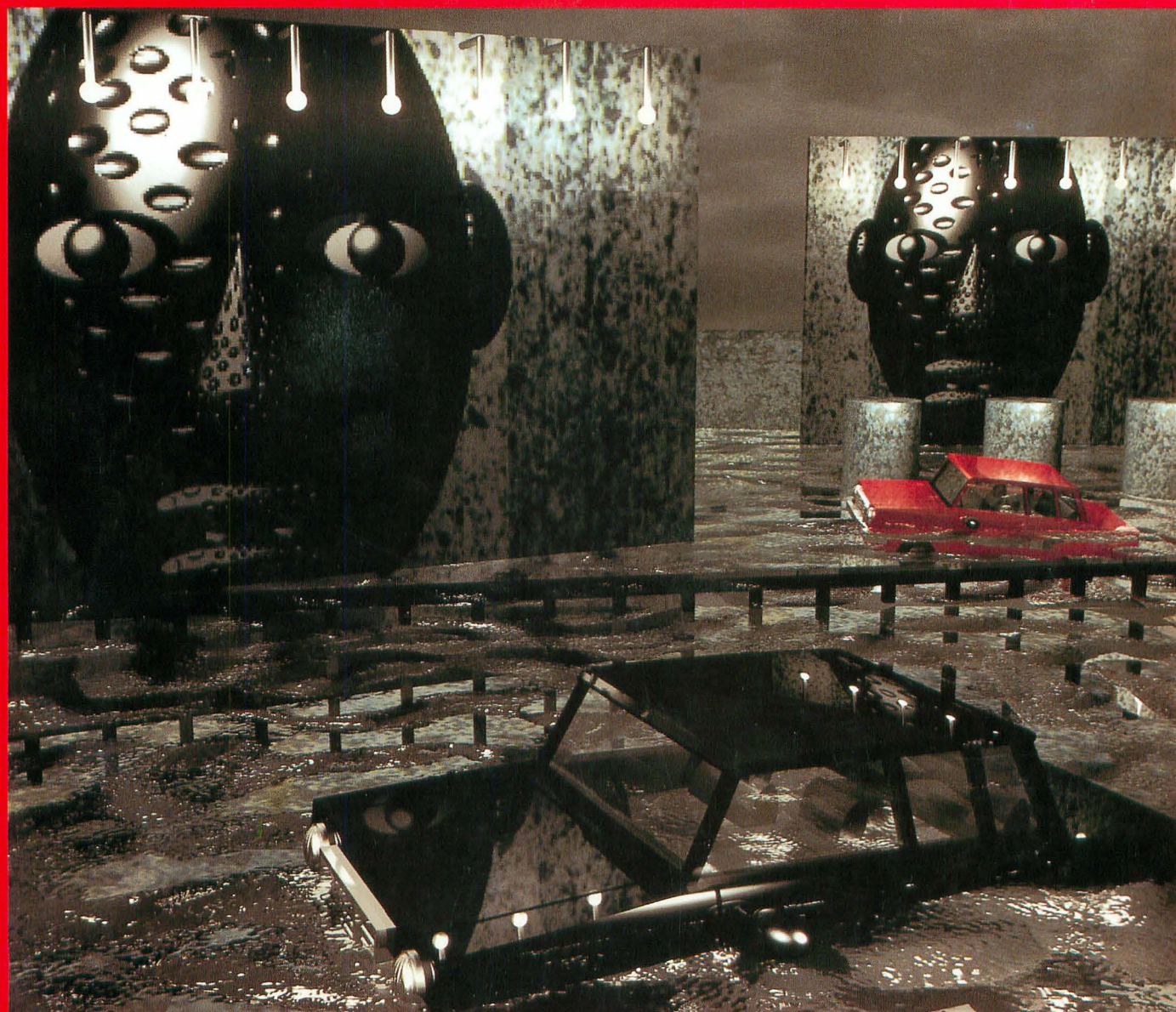
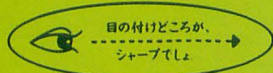
特集 SoundEffects

FM音源効果音のすすめ/FFTを使った畳み込み演算
エフェクタ処理の実際/Z-MUSIC ver.3.0の概要

新連載 ピコピコエンジン活用講座

3
1995

SOFT BANK
オー/エックス
定価680円

目の付けどころが、
シャープでしょ.

■実画面：1,024×1,024ドット、表示画：768×：512ドット

●画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコンなどは、SX-WINDOW ver.3.1がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中の「シャーペン」で表示している文字のフォントはツァイト社の、「書体倶楽部」のフォントを使用しています。

❶「パターンエディタ」で作成したデータを背景に設定可能。

❷日本語フロントプロセッサ ASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❸ESC/Page, LIPSIII, PostScriptに対応したプリンタが利用できます。

❹付属アプリケーション「シャーペン」編集例。文字ごとに文字種・文字の大きさの指定、装飾が可能。またインライン入力をサポート、イメージデータの貼りつけもOK。

❺512×512ドットの範囲内で65,536色の表示が可能。

❻「CGAウィンドウ」、65,536色（最大）のコンピュータアニメーション表示が可能。

❼異なる画像フォーマットへのコンバートが可能。

❽アイコンデータや背景データを作成する「パターンエディタ」。

❾オリジナルに作成したアイコンパターンの例。

❿Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。

■「SX-WINDOW ver3.1システムキット」のお問い合わせは…シャープ㈱電子機器事業本部（液映）システム機器推進プロジェクトチーム　〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161（大代表）へ

フィールドが、膨らむ。

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

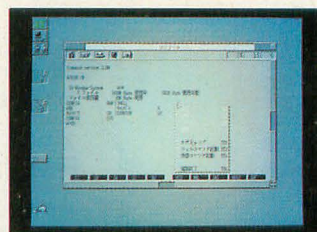
GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。

●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）

●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

SX-WINDOW ver.3.1の
データ利用環境

今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

「SX-WINDOW ver3.1システムキット」CZ-296SS(130mmFD)/CZ-296SSC(90mmFD) 標準価格22,800円(税別)

68買ったら
EXEクラブ
へ入ろう!

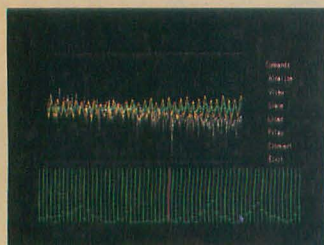
EXE
クラブって
何だ?

X68030/X68000を手に入れて、いろいろチャレンジしたい皆さん。情報のチャンネルは多いほどいいですよね。ということで、EXEクラブは68ユーザーのための水先案内人。あなたのチャレンジを強力にバックアップしますよ。

本体同梱の入会申込ハガキを送るだけで、自動的に無料入会。さらに下記の特典付き。

●Z'sSTAFF、書体倶楽部はツァイト社の商標です。●ESC/Pageはセイコーエプソン株式会社の登録商標です。●PostScriptはアドビシステムズ社の登録商標です。●EGWordは株式会社エルゴソフトの登録商標です。

シャープ株式会社

資料請求券
ペリフェラル
Oh!X
3係

特集 SoundEffects

ディグダグII

VIEW POINT

ピコピコエンジン

(で)のショートプロぱーてぃ

システムX探偵事務所

Oh!X

C O N T

●特集

●カラー紹介

●シリーズ全機種共通システム

●読みもの

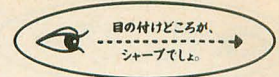
〈スタッフ〉
●編集長/前田 徹 ●副編集長/植木章夫 ●編集/山田純二 高橋恒行 ●協力/有田隆也 中森 章 林 一樹 吉田幸一 華門真人 朝倉祐二 大和 哲 村田敏幸 丹 明彦 三沢和彦 長沢淳博 清瀬栄介 石上達也 柴田 淳 瀧 康史 横内威至 進藤慶到 菊地 功 伊藤雅彦 ●カメラ/杉山和美 ●イラスト/山田晴久 江口響子 高橋哲史 川原由唯 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子 加藤真二 ●校正/グループごじら

表紙絵：塚田　哲也

1995 MAR. 3

E N T S

■広告目次

ビデオグラフィックスの
世界へ。
■お問い合わせは… シャープ株式会社
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
資料請求券
X68030
Oh!X
3係

1,677万色対応、ビデオ映像を高画質・高速取り込み

テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば"ビデオスキャナ"とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

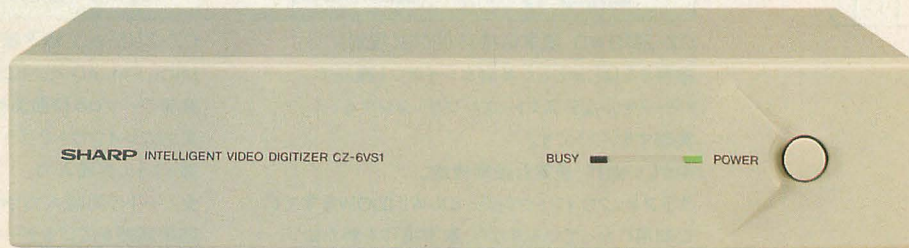
※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。

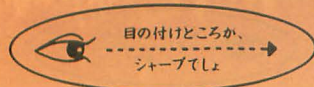
※SX-WINDOW版はバージョン3.0以降(メモリー4MB以上)、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上(メモリー8MB以上)が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

SHARP INTELLIGENT VIDEO DIGITIZER CZ-6VS1　BUSY　POWER

■SCSIインターフェイス採用：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2(FAST)インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2(FAST)対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、Macintosh IIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

標準価格178,000円(税別)

●さらに実用的なウィンドウシステムへの進化

SX-WINDOW ver3.1システムキット

CZ-296SS(130㎜FD)/CZ-296SSC(90㎜FD) 標準価格22,800円(税別)

ASK68K Ver3.0を利用したインライン入力のサポート、Human68k/BASICコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できるコンソールのサポートをはじめ、シャーペン.Xをワープロとして利用できるよう機能アップ。また、さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ドローデータ(FSX)/フォントデータ(IFM)処理の高速化も実現しています。

※コンソールでは、SX-WINDOWと処理が重複するものは実行できません。 (4MB)

●定評のGUI対応ウィンドウワープロ

EGWord SX-68K

CZ-271BWD 標準価格59,800円(税別)

ウィンドウワープロとして評価の高いEGWordのSX-WINDOW対応版。キャラクタベースのワープロを超えたグラフィカルユーザーインターフェイス(GUI)による手軽なDTPソフトとしても優れた表現力を発揮します。定評ある日本語入力方式(EGConvert)によるインライン入力、さまざまなグラフィックデータ(GScript)やテキストデータの貼り込み、また文書互換を実現するEDF(Extended Document Format)形式をサポートしています。 (4MB、ver.2.0)

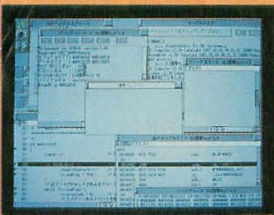
※5MB以上の空きのあるハードディスクが必要です。

●SX-WINDOW開発支援ツール

SX-WINDOW 開発キット Workroom SX-68K

CZ-288LWD 標準価格39,800円(税別)

SX-WINDOW用のソフト開発に必要なツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解が深まる33種(基礎編23種、応用編4種、実用編6種)のサンプルプログラム付き。

※ご使用に当ってはC compiler PRO-68K ver.2.1が必要です。 (4MB、ver.2.0)

●SX-WINDOW開発キットのサポートツール

開発キット用ツール集

CZ-289TWD 標準価格12,800円(税別)

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視・確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。 (4MB、ver.2.0)

●SX-WINDOW対応ドローイングツール

Easydraw SX-68K

CZ-264GWD 標準価格19,800円(税別)

イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは他のSX-WINDOW対応アプリケーションでも利用でき、企画書などの作成をサポート。ページプリンタドライバも標準装備。 (4MB、ver.3.0)

●ウィンドウ対応グラフィックツール

Easypaint SX-68K

CZ-263GWD 標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。 (2MB、ver.1.1)

●SX-WINDOWを楽しく使うためのアクセサリ集

SX-WINDOWデスクアクセサリ集

CZ-290TWD 標準価格14,800円(税別)

SX-WINDOWをさらに便利に楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、スクラップブック、スケジューラ、アドレス帳、電子手帳通信ツール、パズルなど、12種の豊富なアクセサリが収められています。 (4MB、Ver.3.0)

●マルチタスク機能をはじめ通信環境がさらに充実

Communication SX-68K

CZ-272CWD 標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションを実行中でも簡単に通信が可能。自動ログイン機能やプログラム機能、など豊富な機能をサポートしています。 (2MB、ver.1.1)

●FM音源サウンドエディタ

SOUND SX-68K

CZ-275MWD 標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。 (2MB、ver.1.1)

●SX-WINDOW対応になってさらにパワーアップ

倉庫番リベンジ SX-68K ユーザー逆襲編

CZ-293AW(130mmFD)/CZ-293AWC(90mmFD) 各標準価格6,800円(税別)

倉庫番10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。AI機能やエディット機能、キャラクタ変更機能も装備。半年で解けたらあなたは天才?です。 (2MB、ver.1.1)

PRO-68K シリーズ

●X68030/X68000対応

CZ-295LSD 標準価格44,800円(税別)

※メインメモリ2MB以上が必要です。

C compiler PRO-68KのX68030/X68000対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フローティングライブラリを付属しています。

※(2MB,ver.1.1)の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。

●EGWord、EGConvertは株式会社エルゴソフトの登録商標です。

SEGA SATURN MAGAZINE

セガサターンマガジン

NEXT GENERATION SEGAGAME MAGAZINE

540YEN

Ⓒセガ・エンタープライゼス

3月号

2月8日発売!!

[特集]
ゲームメーカーより愛をこめて
他機種に負けないサードパーティー数十社に直撃インタビュー!!

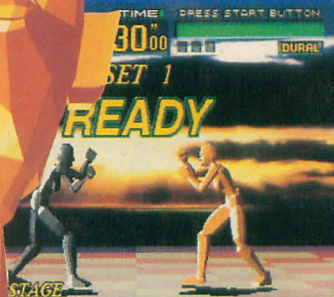
[AM2研EXPRESS NEO]
バーチャファイター2 隠し技大公開!

[SEGASATURN COMPLETE GUIDE]
発売後のセガサターンソフトを徹底攻略！
バーチャファイター ビクトリーゴール

[COMING SOON SOFT]
発売直前！期待のセガサターンソフトを大紹介！
デイトナUSA／パンツァードラグーン／VIRTUAL HYDLIDE
ブルーシード／GOTHA／上海 万里の長城／アイドル雀士スーチーパイSpecial

[NEW RELEASE TITLE]
最新のセガサターンタイトルをキャッチUP！
RAMPO／ダイダロス／グレイテストナイン／輝水晶伝説アスタル

[スーパー32X新作情報]
TEMPO／メタルヘッド

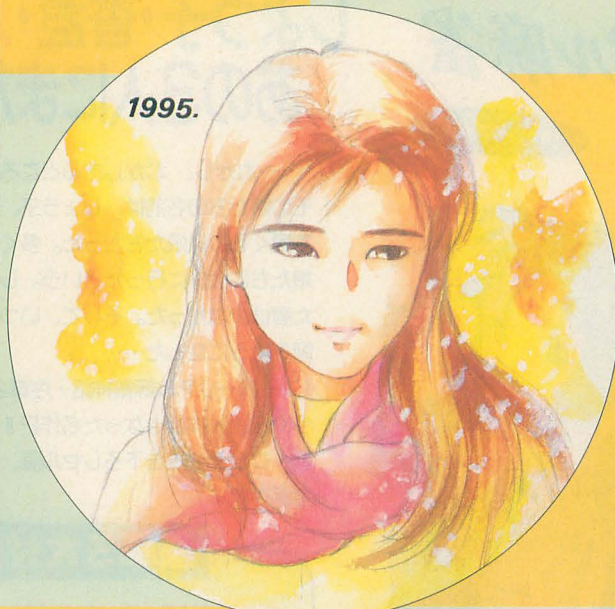
[MEGA-CD・メガドライブ最新情報]
大封神伝／ライトクルセーダー／エイリアンソルジャー／NBA JAMトーナメントエディション／NFLクォーターバッククラブ'95

ゴールド・デュラルも使える！
バーチャファイター裏技完全先取り大公開!!

SOFT BANK お近くの書店でお求め下さい
ソフトバンク株式会社／出版事業部 販売局 TEL.03-5642-8100

The
スーパーファミコン専門情報誌
3/3号
スーパーファミコン
特別価格450円(税込)隔週金曜日発売
全国の書店、コンビニエンスストアにて好評発売中!
ソフトバンク出版事業部
SOFT BANK
特集
(なぜ・なに)33-3÷次世代機編
好評!(なぜ・なに)シリーズの第6弾は今話題の次世代機編だ!!
特報!期待の人気RPG3大作
スクウェア
「クロノ・トリガー」
チュンソフト
「不思議のダンジョン2~風来のシレン」
アトラス
「旧約・女神転生」
2/8
MOON
E
MAG
36
335
魔獣ケルベロスが現れた!
1 ナカジマ
2 レムリアン
3 キャンサー
4 ユミコ
読んで得するスーパーガイド㊢新作SUPER GUIDE
魔神転生II
クラシック・ロードII
最新作をキャッチアップ!新作FRONT LINE
第4次スーパーロボット大戦
ウィザードリィVI~禁断の魔筆
スーパーファミスタ4
ロックマン7~宿命の対決!
実況パワフルプロ野球2
悩んでるタール人を救え!
「三國志IV」武将データ完全紹介
HP 8468/13000
EN 200/200
HP 3200/3200
EN 160/160
ショウ「やるかよっ!」
2大特別付録
「エストポリス伝記II」攻略読本
中綴じスペシャル
「フロントミッション」徹底攻略(前編)

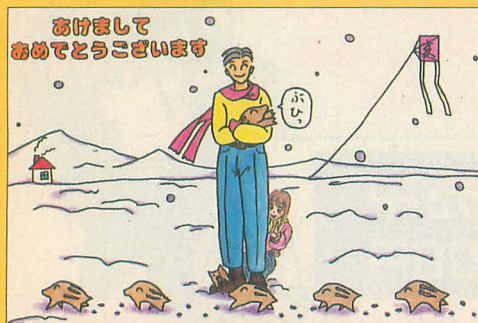
Oh!X reader'sぎゃらりぃ

今年もやってきました年賀状紹介コーナー
力の入ったイラストをどうもありがとうございます
これからもOh!Xをよろしくおねがいしますね

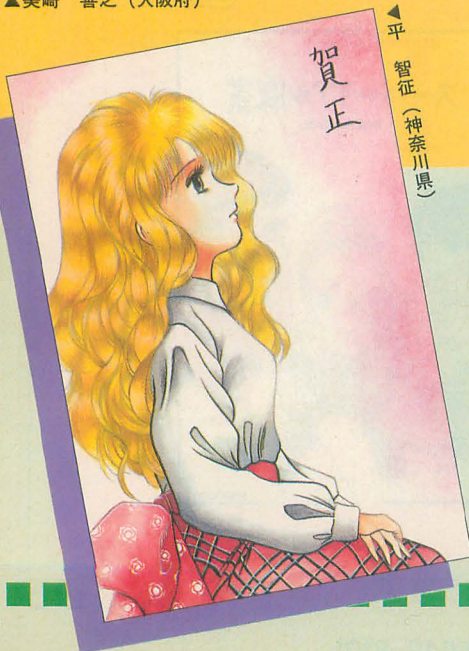
▲山西　孝到（大阪府）

▲前田　基行（兵庫県）

illustration：Y.Kawahara

▲加藤　隆（佐賀県）

▲占部　哲彦（広島県）

▲美崎　善之（大阪府）

▲大高　孝平（宮城県）

▲吉岡　洋明（埼玉県）

◀平　智征（神奈川県）

▶石田　伯仁（神奈川県）

▲島田　貴之（埼玉県）

▲日高　光代（宮崎県）

▲近藤　隆生（埼玉県）

▲濱田　信一（神奈川県）

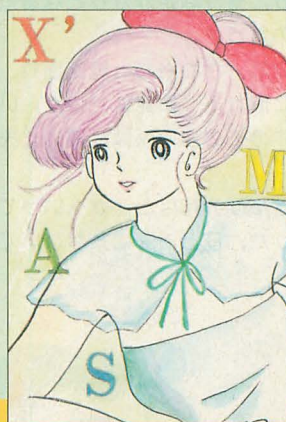
▲清家　亜紀（福岡県）

▲青木　一師（奈良県）

▲八亀　桂一（神奈川県）

▲森井　浩之（兵庫県）

▲今井　健生（奈良県）

▲曽谷　修二（兵庫県）

▲大嶋　靖浩（栃木県）

▲佐藤　貴是（神奈川県）

▶玉野　健一（奈良県）

▲能口　政士（大阪府）

▲岩瀬　貴代美（福岡県）

［第46回］
うどんげ

江口　響子

響子inCGわ〜るど

海から吹く風が，しめった春の匂いを運んできた。昨日までは，冬のつきはなした匂いだったのに……。

浜に出た。波は優しくうねり，朝の太陽に照らされ，つややかな魚の鱗を思わせた。

蜃気楼だろうか。海の面に霞が集まって，ゆらゆらと立ち昇り，徐々に形を成していった。固まって，卵のようになった。

うどんげ……言葉が口をついて出た。

幼い日，いまよりもずっと東京に緑が多かったころ，うどんげの花というのを見たことがある。それは，玄関脇にあった笹の葉の裏側に，十数本固まってくっついていた。直径1ミリほどのごく淡い色の楕円体に，1本の白く細い糸がのびていて，もう片方が，確かな足がかりとなる笹の葉に結ばれているのだった。

風が吹くたびに，さわさわと笹は揺れ，うどんげの花はすぐに取れそうに見えた。が，いかにもはかなげな物体のその白い糸は，案外丈夫だったのである。

数日後，うどんげの花はどこかへいってしまった。白い糸だけが，笹とともにたなびいていた。

うどんげの花とは，クサカゲロウの卵である。孵化して幼虫になり，やがて羽化して1～2センチぐらいの成虫になる。成虫は，淡く透明な緑色をしていて，生きることができるのは，数時間から1日なのだ。

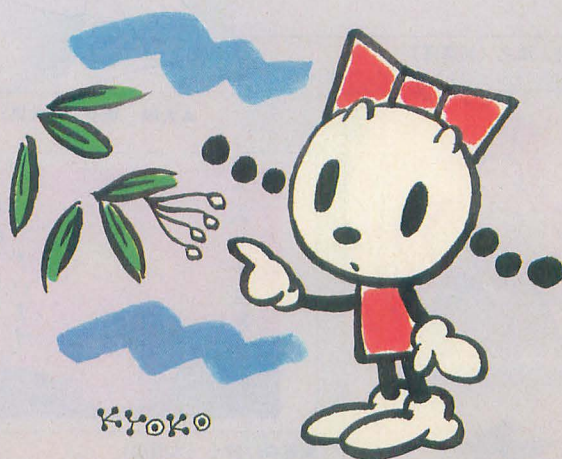
うどんげには，こんな意味もある。インドの想像上の植物～いちじく科で三千年に一度花を咲かせるという……。

「よし，今年こそ登るぞ」

後ろから声がした。

振り向くと，二十歳ぐらいの青年が立っている。脇に長いはしごを抱えていた。彼は，私がいるのに少し驚いた様子だったが，そのまま遠浅の海にずんずんと分け入っていった。ほどなく卵の下に着くと，はしごを立てかけ，登り始めた。しかし，卵は少しづつ薄らいでいき，彼は，はしごの半ばで取り残された。戻ってきたとき，彼はくやしそうにいった。

「もう少し，あとひと息でうまくいったのに。あの上から景色を眺めたら気持ちがいいだろうなあ〜。よし，来年こそ」

1年が過ぎた。吹く風に春の匂いが感じられた

最初の日，私は朝早く浜に出た。暖かな色の海原が広がっていた。

大気が雲のように形をとりはじめ，なめらかな物体に固まっていった。卵の形。去年の春と同じ現象だ。

背後で声がした。

「登るのはやめにしました」

なぜと聞く間もなく，彼がいった。

「実際にはしごをかけなくても，上に登ることができたからです。遠くまで見渡せて，とても気持ちよかった……」

はじめ，よく飲み込めなかった。が，脳裏にある，はしごを立てかけた卵の映像が，うどんげの花と重なったとき，突如として理解した。

彼は，自らの思惟の世界で，想像力をもってして，そのことを成し得たのだ。彼の白い糸は風になびきながらも確かなもので，たとえ卵が消えてしまっても，残るに違いなかった。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを4×5ポジで出力
作成手順
X68000SUPERで卵の画像とそのαチャネルを作成。使用ソフトは，サイクロン。MacintoshQuadra840AVで背景をレイトレーシングで作成。使用ソフトは，KPT Bryce.。Photoshop2.5Jで背景のPICTファイルをRGBファイルに変換。XIN/XOUTⅢ（電机本舗）で，X68000に転送。卵の画像と背景をαチャネルを使って，X68000上で合成。

Oh!X Graphic Gallery

XL/Imageテストレンダリング

今月のGraphic Galleryでは1994年11月号で掲載した森山氏の恐竜と，文月氏の「TORNADE」のデータをXL/Imageでレンダリングしたものを紹介します。

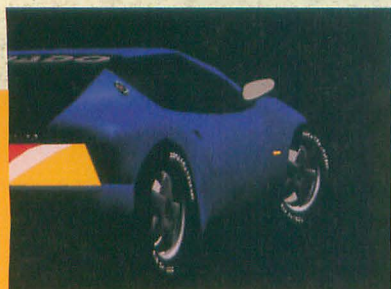
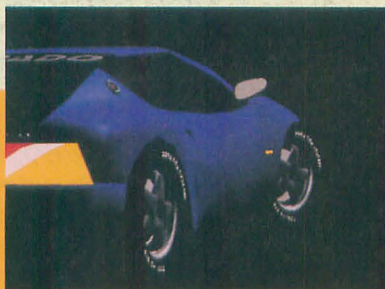
写真1　11月号の恐竜のデータ

写真2　XL/Imageのデータ形式にコンバート

写真3　環境マッピングを指定して作画

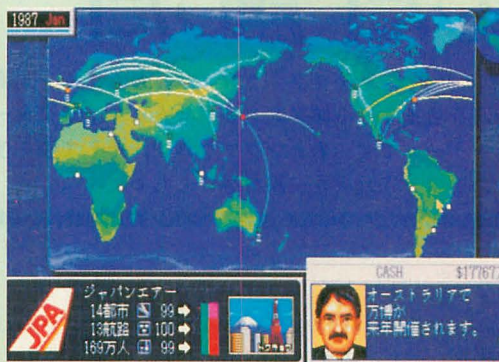
写真4　reflectionをONにして影をつけた

●写真1〜4のようにして作成した画像のアニメーション。金属の質感をもった恐竜が走る(本来は15フレームのデータで，「電脳倶楽部」の3月号に収録されている)

これは文月さんの「TORNADE」ですが，XL/Imageでレンダリングしてあります。DōGAのレンダリングと比べてみましょう。

DōGAの普通のレンダリング

DōGAのディザリングのレンダリング

XL/Imageのレンダリング

SOFTWARE INFORMATION

気になる新作「プリンセスメーカー」「地球防衛MIRACLE FORCE」は，現在も鋭意制作中。来月には新しい情報が届けられる……かな。また，新作情報でかなり変化がありました。要チェックです。

TAKERU名作文庫ソフト

1994年2月より実施されている，過去の名作たちをお手頃価格で販売する「名作文庫ソフト」シリーズに，新たな作品が加わった。タイトルと価格は以下のとおり（価格はすべて税込）。（ ）内はパッケージ販売時の価格である。

光栄

タイトル	価格
太閤立志伝	3,400円（9,800円）
エアーマネジメント	4,100円（11,800円）
ヨーロッパ戦線	4,500円（12,800円）
伊忍道　打倒信長	3,400円（9,800円）
ロイヤルブラッド	2,700円（7,800円）
蒼き狼と白き牝鹿・元朝秘史	3,400円（9,800円）

ビング

タイトル	価格
スーパーリアル麻雀PII＆PIII	8,800円（12,800円）

これらのソフトはすでに1994年11月下旬までに発売済み。また，3月下旬には新しいメフトたちがラインナップに加わるようだ。

X68000用　3.5/5″2HD版
TAKERU　☎052(824)2493

発売中のソフト

★VIEW POINT　ネクサスインターラクト　1/27
X68000用　5″2HD版　7,800円(税別)

新作情報

★ディグダグ/ディグダグII　電波新聞社　2/24
X68000用　5″2HD版　5,300円(税別)

★X CASE　Beシステム
X68000用　5″2HD版　19,800円(税込)

★Traum　象スタジオ
X68000用　5″2HD版　価格未定

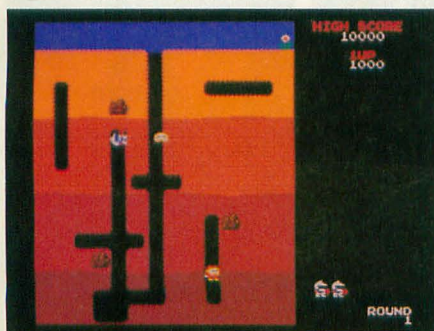
★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)

★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版　価格未定

★プリンセスメーカー　ニュー
X68000用　5″2HD版　14,800円(税別)

★フォント＆ロゴ デザインツール 書家万流SX-68K　シャープ
X68000用　3.5/5″2HD版　価格未定

●

いままで，新作情報として掲載していた「ロボスポーツ」「鮫！ 鮫！ 鮫！」「達人」「エアバスター」「サバッシュII」は，すでに制作が中止されていましたので，新作リストから外させていただきました。「スタークルーザーII」については，移植が具体的に明らかになった時点で情報を掲載いたします。情報の確認が遅れたためご迷惑をおかけした関係者各位，読者の方々におわびいたします。

掘った! 落とした! 崩した!

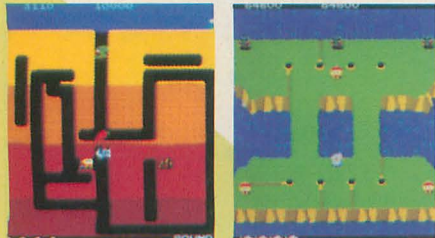
Yaegaki Nachi
八重垣 那智

「ディグダグ」「ディグダグII」は，ともに敵を誘導し一気全滅を目指す，戦略要素を多く含んだキャラクターアクションゲームです。岩とプクプクボン，そして頭を使ってプーカとファイガーを一網打尽にしちゃえ。

友人と話していて唐突に，十年前になにをしていたか，などという話題になることがある。しかし，人間の記憶なんて曖昧なもので，現在から順にたどっていくと，どんどん不鮮明になり，記憶の輪郭や色がボロボロと壊れていき，結局はすぐに思い出せないことが多い。

こういうとき私は，その年に出たゲーム，しかもヒットしたゲームを思い出すことにしている。さすがにゲームの名前を聞いただけで，当時のヒットソングや総理大臣の名前まで連想するのは無理だが，自分がなにをしていたのか程度なら思い出すことができる。とはいえ，あの階段の狭い地下のゲームセンターはいまどうなっているのだろうか，ということまで考え出すと，どうにも気になってしまい，いてもたってもいられなくなって本当に見に行ってしまったりするので，この方法に頼るのは，ほどほどにしておいたほうがいいようだ。

もっと深い穴を

もうすでにお馴染みとなった，ビデオゲームアンソロジーの12作目である。今回は，「ディグダグ」「ディグダグII」という親子ものカップリングで登場することになった。どちらもナムコのキャラクターゲームであるが，ナムコものとしては初のカップリングで登場である。どちらも戦略的要素が追求された奥の深さが評価されたゲームで，ちょっとマニアックなところがポイントになっている。とりあえず順番に見ていくことにしよう。

まずは元祖である「ディグダグ」のほうであるが，1982年の初頭にデビューしたゲームで，地中に棲むモンスターを自分で穴を掘り(作り)ながら退治するゲームである。赤い風船のようなプーカと，恐竜のような緑のファイガーを画面上から全滅させれば1面クリアとなる。敵を倒す方法は2種類あり，自分の正面にモリを投げて敵を刺し，ボタンで空気を送ってパンクさせるプクプクボン(公式にこういう名前)方式と，真下を掘ると落下する岩に敵を潰させる岩石落とし方式がある。どちらの方法にしても敵を倒すためには，自分で穴を掘り，道を切り開くことになるので，自分の動きがゲームの展開を大きく左右しているところに注目したい。「ディグダグ」は1にも2にも掘るゲームなのである。

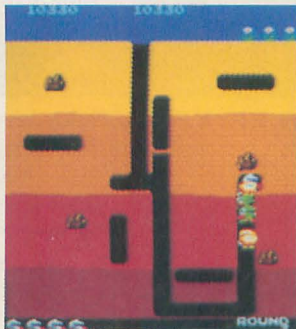
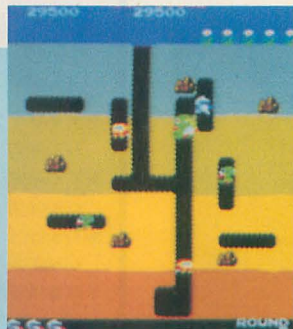
もっと長い穴を

こういったプレイヤーによる展開の幅広さが，戦略的穴掘りゲームといわれるところであり，実態は見た目からは想像しづらいパターン性の高いキャラクターゲームであることがわかる。特に岩を使って敵を潰す場合，まとめて潰すことで高得点になるので，できる限り敵を集め，一網打尽(一岩打尽か?)にする方法を探し始めることになる。こうなると際限なくハマってくのは間違いないといえるだろう。

こういったボーナス得点目当てのプレイを狙っていくと，ゲームが様相を変え，最初のステージから頭を使うハメになるのは，ナムコの黄金期(「ギャラガ」から「パックランド」まで)のゲームに見られる素晴らしい特徴である。敵キャラクターも2種類しかなく，かなり地味なゲームではあるが，敵の配置や地形との組み合わせで，意外な奥深さを醸し出している点は，キャラクターアクションゲームのお手本といえる。ちなみに，これらのキャラクターはナムコのキャラクター商品の元祖にもなっていて，縫いぐるみや風船，帽子など（いまではありふれているが）当時としては珍しい多角的な展開がなされていたことも興味深い点だといえるだろう。

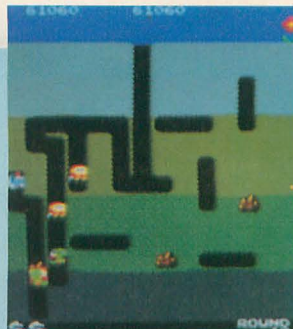
X68000版をプレイした感じでは，見た目の違いはほとんどわからなかった。目変化のタイミングなど，細かいところまでいわれると自信がないが，かなり研究してあるようだ。また，このゲームにはいくつか有名なテクニックがあり，ナムコから公式に豆本という形で提供されているが，こういった資料に載っているようなものは，チェックした限りでは大丈夫だった。

岩の下を掘ってから落ちるまでの間に，振り向いて敵を打って麻痺させ，自分は逃げるというかなり高度なテクニックであるパラライザー打ちもできたので，基本的に問題はないだろう。「ディグダグ」の世界を味わうには十分な移植の出来ではないだろうか。

X68000用　5"2HD 5,300円（税別）
電波新聞社　☎03(3445)6111

敵を誘導して一網打尽!

これがいわゆる病気掘り

大きなお花が咲きました

もっと大きく崩せ

あとは下方向にヒビを入れるだけ　目指せ，一気８万点！　……無念。やられてしまった

もう１本のほうの「ディグダグII」であるが，これは1985年の春にリリースされ「ディグダグ」の続編と銘打っている。しかしゲーム内容は一変してしまっており，まったくの別ゲームといってもいいだろう。ただし，キャラクターが前作とまったく同じであるためか，続編というイメージにそれほど違和感は感じられないようになっているので，細かいことはいわないほうがいいだろう。

舞台は地中から一転して平面である海上の島に変わり，その島の上で同じくプーカとファイガーを退治するのが目的となっている。島は面によってその形がさまざまに変化し，最大で画面横２画面分ほどにもなる。画面からはみ出た場合は，左右のスクロールを使って表示されるので，固定画面だった前作との最大の違いになっている。しかも，この地形のバリエーションの豊富さが，前作と同じ，たった２種類の敵キャラクターしか出てこないゲームにおいて，変化と演出を大きく左右しているところにチェックが必要だ。

舞台が変わっても敵の倒し方は相変わらずのプクプクポンと，今回は岩がないために新たに編み出された島崩しという大技である。直接的には新設されたドリルボタンを島のあちこちにある杭の上で使うことで，地面にヒビを入れるだけなのだが，そのヒビがつながり島が分断されると面積の少ないほうが陥没するようになっている。これが島崩しであり，崩れた地面の上に敵がいれば，その敵を退治することができる。

もっと派手に崩せ

もちろん敵を集めて一気に崩せば最高８万点の高得点ボーナスになるので，どんどん狙っていくほうがいいだろう。「ディグダグII」でもやはり，これがゲームの醍醐味になっているのである。しかも面ごとに地形が異なるため，プレイヤーは地形に応じた敵の誘導と，有効な島の崩し方を考えなくてはならなくなっている。前作よりもわかりやすくパターン性をアピールした演出が光っているといえるだろう。

そういった意味では敵のキャラクターをあまり意識するというよりも，どちらかというと地形を相手にしているような意識がプレイの際に強く出てしまうキライがある。そのせいか，前作よりもアクション性は上がっていても，キャラクター的には弱い印象があり，当時の流行から考えると見てくれの悪いゲームだったといえなくもないだろう。結果としてそれほどヒットしなかったのは，そういった部分によるところが大きいのではないだろうか。

しかし，「ディグダグII」は最初の面から無理をして高得点を狙い，自ら危険に身を晒してスリルを味わうナムコゲームの趣は衰えておらず，ある程度腰を落ち着けて攻略しなければ先の面を見ることができない厳しさも備えている。X68000版でもそれは同じで，攻略も同じように使えるし，敵の誘導といった感覚的な部分についても，前作同様に合格といっていいだろう。タイミングなどを意識した小手先のテクニックといったものはかなり減っているので，アクション的な部分で爽快感を味わえるようになっている。名作ではないが，小粒にまとまっているゲームなので，この機会にじっくりプレイしてみることをお勧めしておこう。

もっと味わい深く

この２つのゲーム，オリジナルはどちらも縦画面のゲームだが，上下に各種ステータスの表示などがあるため，実際プレイする画面はそれほど縦長というわけではない。そのため画面には，ほとんど感覚的な問題はないと思われる。「ディグダグ」ではスコアや残機を画面端に出すことで，「ディグダグII」ではゲームに関係ない海の部分をカットすることで，画面の印象を変えないようにしている。

しかも，無保証の領域ながら特殊な縦長の画面表示サイズに切り替え，本物そのままの画面レイアウトでプレイできる機能もついている。こういった細かな工夫や気配りは，移植ソフトの違いに敏感な人にはありがたいし，そこまでこだわる中身の確かさといったものにも安心感が生まれるので，大いに歓迎したい。特殊な画面モードの恩恵が受けられない人も多いかもしれないが，あくまでもオリジナルを忠実に再現しようという姿勢は決してマイナスにはならないだろう。

また，相変わらずアレンジやおまけ機能もないのだが，これはアンソロジーシリーズのポリシーであると考えることにする。なにも足さず，なにも引かない潔さと純粋さは移植ゲームのひとつのあるべき姿といえるからである。

最近少し寂しくなったX68000のゲームソフトだが，こういった古きよきものをできる限り残していこうという流れや，マシンの限界を求めていこうという姿勢がある限り，まだまだがんばれるという印象を受ける。もっとX68000自体を見直すような移植ソフトも，ちょっと期待してみたいものである。

ワナを絞る快感

とにかくこの２つのゲームの特徴は，敵に罠をしかけて一網打尽にする快感に尽きます。ギリギリまで追い詰められても，ニヤリとしながら最後の一手を下して一発大逆転する醍醐味は，なかなかほかのゲームでは味わえないものでしょう。あと「ディグダグII」には，一部地形の違ったニューバージョンというのがあるのですが，個人的にはそっちのほうが好きなので，両方対応にしてほしかったところです。ちょっと残念。どちらにしても，腕が錆びついていてロクにプレイできないので，気合を入れてリハビリしないとダメかなぁ，うんうん。

ディグダグ

総評	0　5　10
ゲーム性	★★★★★★★★
グラフィック	★★★★★★
技術	★★★★★★
サウンド	★★★★★
戦略性	★★★★★★★★

ディグダグII

総評	0　5　10
ゲーム性	★★★★★★★★
グラフィック	★★★★★★
技術	★★★★★★
サウンド	★★★★★★
戦略性	★★★★★★★

幻の名作「VIEW POINT」は……

Nishikawa Zenji
西川 善司

NEO・GEOで密かな名作といわれている「VIEW POINT」がX68000に登場した。ちょっと待たされてしまったが，オリジナル版独特のグラフィック，音楽，ゲーム性がどれだけ忠実に再現されているかを見ていこう。

今回，ネクサスインタークラクトよりX68000に移植された「VIEW POINT」のオリジナルは，1992年発売のNEO・GEO用ソフトだ。クォータービュースクロールシューティングという変わった作風で，当時，「空へ舞い上がる無敵対空技をもった格闘ゲーム以外はゲームにあらず」という雰囲気の漂った日本のアーケードゲーム界に颯爽と登場した。

ポリゴンでレンダリングして描かれたと思われる妙に硬質感のある立体的なグラフィック，そして生物的な動きを見せる独創的なデザインの敵キャラ……，画一的なものばかりが揃ってしまった当時のシューティング界の中においても，かなり際立った存在であった。

NEO・GEOというと最近では「NEO・GEO CD」が有名で，これに対応した過去の名作ソフトが安くCD版となって再販されている。しかし，この「VIEW POINT」はいまだそのラインナップにも入っていない。名実ともに幻のソフトなのだ。

ついにX68000版登場

8方向レバー，2ボタン(ショット，ボンバー)のオーソドックスな操作性で，出てきた敵を倒すという単純明快なゲーム内容。ショットは押し続けるとチャージされ，いわゆる波動砲も撃つことができる。

X68000用　5″2HD版　7,800円（税別）
ネクサスインターラクト　☎03(5474)3581

自機のパワーアップは，左右にオプションがつく以外特になし。ときどきシールドアイテムが出てくるが，敵の攻撃がハンパじゃないのですぐ外れる。ボンバーも途中で補給できるが，ストックはたった3つなので，これまた厳しい。各ステージ最終地点にはボスが待ちかまえているが，凶悪なまでに強くて堅い。もう泣き出したくなるような猛攻を耐えて耐えて耐え抜いていかなくてはならない，結構マゾヒズムなゲームなのだ。この猛攻が気持ちよく思えてきたら，君はもうビューポインターだ。

X68000版は家庭用NEO・GEO版と違って1ミスしてもその場から再スタートなのが，せめてもの「地獄で仏」的な救済。難易度設定や残機設定もあるし，初心者でもステージ1,2くらいまではクリアできると思う。その先はただ鍛練あるのみ。そうそう，クォータービューということで，自機の当たり判定がわかりづらいかもしれない。そんな場合は，キャラクターのカゲに当たり判定があると思ってプレイすると，かなり遊びやすくなるぞ。

また，パソコンではすでにFM TOWNS版が登場しているがこれがなかなかの出来。ゲーム性はもちろん，グラフィック，サウンドともにかなりのクオリティで再現されていた。で，X68000版はというと，実に残念だが結論からいえば，頭に「Tiny」をつけたほうが世の中丸く収まるものよウワッハッハといった感じの出来である。

グラフィック

256×256の画面なのはまぁしかたない。オリジナルと画面の比率が異なってしまい，通常のディスプレイの垂直振幅ではゲーム画面がやけにつぶれた感じになってしまっているのも我慢しよう(ディスプレイのツマミを調整すればOKだし)。

NEO・GEO版のグラフィックをそのまま吸い出してきた感じなので，動画落ち(アニメパターンが容量節約のために簡略化されること)のような凶悪な手抜きはない。しかし，垂直帰線期間を見ずに描き換えを行っているため，動画が無意味にちらつくことが多い。スプライトの最大表示数オーバーのちらつきとは違ったちらつきが目につく。このへんはゲーム作りの基本なのでもっともっと精進されよ，陽一君！　といったところ。タイトル画面でのたった1機の自機の回転アニメがちらついているのはあまりにも恥ずかしい。リアルタイムにレンダリングしているのなら話は別だが。

あとハードの制約か，はたまた技術不足か微妙なところだが（ディスク容量の不足という話も……），背景動画および巨大キャラがかなり省略されている。「ボクちゃんのX68000はこんなものじゃないやい」と悔し涙を流している熱血X68000ゲームプログラマは，世の中に30人くらいはいると思う。たとえば1面の観覧車クルクル攻撃部隊の「カバード」が出現しない。X68000版では，

あ，BGMがスローテンポに……

このくらいの弾数でビビるな！

彼らの出現場所であった丸い広場は遊ぶものもなく寒い風が吹き抜けるだけで，ただただ無情にスクロールアウトしていく。1面の見せ場なのに悲しい。

2面の最初の見せ場。タイトーの「サイバリオン」から飛び出してきたような中ボス(?)のお邪魔クネクネドラゴン「ハイドラ」君は出演拒否。海面のラスタースクロールはカット。うがー。

3面の最初の難関。異なる周期で開閉するゲートをかい潜り……のはずがカット。ゲートの開閉ですり減った土地の描き込みだけが空しく自機の下を通りすぎる。続いて本来なら，地面と区別のつかないようなカムフラージュを施された秘密発進基地のシャッターが突然開き，敵機来襲！　のはずなのだが，これまたこのシャッターがなくなっており，「我々はここにいます，ドモ」と，むき出しの基地の上で敵機は初めから自分の存在を暴露している。一瞬，X68000版の敵軍は軍事予算が削減され，敵基地は未完成……などというバックストーリーが設定されたのかと錯覚したが，そうではないようだ。

このほか洞窟の屋根がないとか，巨大イモムシファミリーの大行進におとうさんの姿はあってもおかあさんと子供が1人いなくなってるとか，ボスの巨大な蛾の羽が羽ばたかず，まわりをうろつく幼虫君が登場しないとか。3面は随分と省略が目立つ。ちなみにこのシーンがパッケージイラストになっているのだがイラストには幼虫君がちゃんといる。ズガーン！

細かいところまで挙げるとキリがないが，4面ではあの名物，風呂桶型シャトル君のお邪魔大ダンスパーティ攻撃が初めから最後までカットされていて，シャトル君自体出てこない。このシーンではなにも出現してこない背景スクロールオンリーのシュールな時間をボス登場まで強いられる。おーい。誰かいないの。

サウンド，BGM

前述したようにこのゲームには斬新なハウスミュージックが採用されていた。音ネタバリバリのアシッド系のものから，オルガンのアドリブメロディが心地よいクラブサウンドまで，まさにハウスミュージックのごった煮，玉手箱，集大成ともいえたあのオリジナルサウンドをどう再現してくれるのか，はたまた再現できるのか，とファンの間ではいろいろと議論が交わされてきたが，結論からいえば，いま現在，私は「最近はだいぶ暖かくなりましたね」とごまかしたくなるような心境だ。

ちょっとうっとおしい特攻トビウオ部隊

4面から敵の攻撃はどんどんエキサイトしていく

まず，残念なことにBGMのリズムパートに，AD PCM音源を採用せずにすべてFM音源でやってしまっている。ときどきAD PCM音のパートが入ることもあるが，ほんとにまれ。「VIEW POINT」が起動中，X68000のAD PCM音源はかなりお休みをいただけるって感じ。さらに原曲でPCM音の集まりだった曲を無理矢理FM音源で鳴らしているため，なんの曲かわからないほど変貌を遂げてしまっている。

でも，MIDIには対応しているんでしょー？　と私は手に汗をかきながらゲームの設定メニューをまさぐったが，そういったメニューはなかった。ガーン。

で，シューティングゲームの命といってもよい効果音だが，ゲームはチープなBGMをバックになぜかほとんど無音で進行する。ただし自機のショット音だけはやかましいくらいに聞こえるが。なるほど，ここは宇宙空間で真空だから外の音は聞こえない，コクピットに乗っている設定の自分は自機のショット音だけが聞こえるってわけね，無理矢理こじつけたのだが，涙はどんどん目の奥から出てくるのであった。

ゲームそのもの

ここまできていうのもなんだが，ゲームそのものの出来はよくない。ドドーン。

まず，やたら重くなる。X68030でも音楽のテンポが遅くなるほど重くなる。音楽のテンポが遅くなるのはキャラクタパターンの転送時に割り込みを禁止してしまっているのが原因なのだろうが，それがわかったところで私にはどうしようもない。ちなみに16MHzだと瞬きしても画面が描き換わらない超スローモーション体験ができ，さらに10MHz機では瞬きしたあと，目薬がさせ，さらに飴を口に放り込むくらいの余裕あるゲームプレイができると思われる。

さらに敵弾が常時点滅していて見にくい。これで表示キャラが増えてくると重くなるだけでなく，自機も姿を隠すので，心臓の悪い人は保護者同伴でプレイしていただきたい。いまさらだがハードディスクのインストールには対応している。

最後に

一応今回レビューしたものは，1月23日現在の開発途中バージョンである。なんとか製品版ではいまの16倍は出来がよくなってもらわないと，かなりまいっちんぐマチコ先生な状態だ。このまま発売されてしまった暁には開発者全員，永平寺で雑巾掛けでもしていただきたいと切に願うものなりなりなりん。ゴーン。

過去の栄光は……

いまさらながらコナミや，SPSの技術というものはものすごかったんだなと思う。たとえば，SPSの移植した同じクォータービューのゲームの「メルヘンメイズ」(ナムコ)を思い出してもらいたい。あれだけ大きなキャラクタがわんさか出てきてもめったに重くならないし，ちらつかない。もちろん10MHz機での話。コナミの「グラディウスII」や「出たな!! ツインビー」も同様だ。

それに比べて今回の「VIEW POINT」はあまりにも志が低すぎ，ファンとしては苦言を呈さざるをえない。X68000は確かに8年前に設計された，いまではちょっと古めのマシンではある。が，スプライトの敵が数機出てきてスプライトの弾を数発吐くだけで処理速度が遅くなるほど，ゲームに不向きなハードでは決してない。今後もX68000用のゲームを開発する予定があるのなら「完全移植」なんて贅沢はいわないから「ちゃんと遊べるもの」を作っていただきたい。また，今回の「VIEW POINT」は，ユーザーにバージョンアップサービスなどの配慮がほしいものだ。

総合評価	0　5　10
移植完成度	★★★
グラフィック	★★★★★★★
サウンド	★★★
技術	★★
難易度	★★★★★★★
熱中度	★★★

THE SOFTOUCH

特別編8

燃えよフェイロン! 魅惑の5HIT COMBO

Fei-Long

Kiyose Eisuke 清瀬 栄介

小学生の頃にブルース・リー・ブームを迎えた世代にとって，ブルース・リーは最高最強の格闘家だ。雑誌の通信販売にはヌンチャク通信講座があり，みんな鼻血を出したら手に取ってなめて，「ペッ」と吐くのがお約束だった（違う？）。

そういう世代のこのボクに，このフェイロンの登場である。この顔，この声，この衣装。これぞ最強の格闘家の誕生である。これでキミもボクもバーチャル・ブルース・リー。ジャッキーもサラもしょせん私の弟子だ，わははははは。

これがフェイロン基本セットだ

というわけで，春麗からフェイロンに乗り換えたボクだが，またしても飛び道具がなかったりする。春麗はスーパーから気功拳を出すようになったのがクヤシイ。お気に入りのキャラが，いっつも飛び道具が使えないのはなぜだっ！

でも「格闘」ゲームなのに，そういう怪しいものを飛ばすほうが邪道なのさ。やっぱり格闘家たるもの，拳で相手を倒さなきゃ。実際フェイロンは飛び道具がなくても，動きは速いしリーチは変幻自在だからがんばればかなり勝てる。

自分も相手も立った状態で頼りになる技は以下のとおり。

●しゃがみ大パンチまたは中パンチ

基本中の基本。立ったときからは予想できないほどリーチが長い。これを覚えておくと，波動拳など飛び道具の出かかりを潰すことができる。大パンチのスキの大きさが怖いときは中パンチでもいい。

●レバー左右＋大キック

近距離で当てると2 HIT COMBOになる蹴り。いきなり出しても当たる確率は高くないので，普段から移動に使ってみたり，しゃがみパンチなどからつなげて使おう。

●烈火拳

声と動きがもっともブルース・リーっぽい技。いきなり入れるのは難しいので，ほかの技からつなぐか，普段からカラ打ちして相手を惑わしておく。それから至近距離では使わないように。拳がすり抜けることがあるのだ。

1発目が入ったら残り2発をすかさず叩き込む。万一ガードされたときは，2発＋紫炎脚もしくは投げというのがビューティフル。3発目を出してしまうと，余韻にひたっているところを餌食にされる。

次に相手に跳び込まれたときの攻撃方法だ。紫炎脚と一緒に「あちょー！」とか叫んで相手をいやーな気持ちにさせてあげよう。

●紫炎脚

昇龍拳タイプの技だが，横方向の移動が小さいので，上にかぶさるようにきた敵に使おう。リュウ，ケンから乗り換えた人は思わずこれを出してしまい，打ち上げ花火と化すことが多いので、見た目はハデだけど，そんなにオイシイ技じゃない。

●大キック

相手が遠めに跳び込んできたときはこれで対処する。クツを相手に当てるつもりで。

これがうわさのバックブリーカーだ（大嘘）

よし，ピヨリ確定。あとは5HITでとどめを刺せ

いやっほう！ 5HIT COMBOだぜ

この蹴りを見てるとわかるけど，フェイロンって案外足が短いのね。同期のディージェイはあんなに足が伸びるのに。

最後にジャンプ攻撃を紹介しよう。フェイロンはクセのある動きをするので，跳び込んで勝てる確率は高い。

●跳び込み大パンチ＋立ち中パンチ

おサルさんでも入る2 HIT COMBOだ。ボクは最初，これを愛用していた。投げ返されることがあるのがツライとこ。

●跳び込み大キック

使っている本人も当たり判定がよくわかっていない謎のキック。攻撃判定が前に広いらしく，相手がつられて跳び上がると勝てる可能性がますます高くなる。

5HITドラゴンへの道

そしてなんといってもマスターしたいのが，跳び込みからのコンボだ。

●跳び込み大パンチ＋大パンチ＋烈火拳3発

これがアツい。大パンチキャンセルからの烈火拳3連発だ。大パンチのあとの烈火拳の入力が死ぬほど大変だが，練習するだけの価値はある。3発目でKOするとナルシスト度100%である。逆に，コンボの途中で相手をKOできるときは，余分な烈火拳はつつしもう。形にこだわる，これもまたひとつのフェイロン道である。

Xになってからは多少牙をぬかれた感のあるフェイロン。やっぱり爽快感ならこのスーパーが一番だ。君もドラゴンを目指そう。アチョー!!

特別編9

やっぱ使うなら主人公でしょ

Miyata Masaaki 宮田 雅章

Ryu

今回から，かなり主人公らしくなったリュウくん（31歳，独身）。それまで歳相応と思われた不精ヒゲもきっちりと剃り，整形を受け（たかどうかは知りませんが），見事に若返りました。初代からリュウを使い続けていた僕にとって，思わず「おのれは何者じゃ」と嘆きたくなる変貌のしようですが，グラフィックのレベルも上がっているので，まあヨシとしましょう。

ファイヤー！ やったなぁ？

スーパーに移行するにあたり，ファイヤー波動拳を身につけました。こいつの使いどころですが，近距離で相手が転ぶことを利用し，連続攻撃の最後にもっていくと，余計な反撃をくわずにすむので，終始ペースを握り続けることができます。

竜巻や昇龍拳が連続攻撃に組み込みにくいことを考えれば，結構重宝します。

パーフェクトで勝とう

リュウに基本戦法というものは存在しません。それだけフレキシブルなキャラだということになるわけですが，それだそのぶん頭と腕が要求されます。使いこなせるようになれば，中間距離では相手を寄せつけず，接近戦では反撃の隙を与えない戦い方ができるので，相当な強さになります。

と，いうわけでリュウを使ううえでの注意点についてのみ触れておきます。

なにも考えずにただ跳び込むだけでは，確実に撃ち落とされます。波動拳も，ただ撃つだけでは連続攻撃の格好の標的です。

そこで，通常技による牽制を行い，跳ばせたり跳び込んだり，という行動をするわけですが，地上戦のメインとなるのは，パンチ系よりもリーチのあるキック系の技になります。しかし，これもただ出すだけでは，まず反撃をくらいます。ですから，当たったらキャンセルをかけて波動拳を撃つなどの事後処理（？）が必要です。

では，こちらよりも足払いの強いキャラがいたらどうするのか，それには隙を見つけて波動拳か，伸びた足に昇龍拳を当てるくらいしか道はありません。傾向としては，足払いの強いキャラに対しては波動拳をうまく使うことで，どうにかなります。

通常技

どれも使いでがあるものばかりです。新キャラほど便利なものが揃っているわけではありませんが，「ホラ，そこの波動拳と昇龍拳しか知らない君。もっと通常技を使いなさい」といいたいぐらい便利です。ほとんどの技にキャンセルがかかるので，リュウの通常技の使いやすさは旧キャラのなかでもトップクラスといえます。

でっきるかな？

腕に自信のある方は，以下の連続攻撃を試してみてください。

●空中竜巻（ガードさせる）→中足払い→波動拳

ほとんど遊び技ですが，くらったほうは戸惑うので，思ったより，有効な技かもしれません。

出すならやっぱり3HIT COMBO 4,500点

●ジャンプ強パンチ→フック→昇龍拳

今回からは，リュウ，春麗以外のほぼ全キャラに入ります。フックの代わりにアッパーが入るキャラもいますので，試してみてください。また，どうしても昇龍拳が入らない，というときはダメージ的にも大差ない強竜巻を使うといいでしょう。

●波動拳→中足払い→波動拳

今回の目玉で，これが入ると気絶確定。やり方ですが，まずは投げや足払いで相手を転ばせます。次に密着状態から，相手の起き上がりに合わせ，後頭部に当たるように波動拳（ファイヤー不可）を撃ち，当たったらばすぐに中足払いを入れます。それにキャンセルをかけ波動拳。この技は3ヒットで稼げる最高得点の4,500点が出ますので，できたらちょっとした自慢になります。ただし実戦での利用価値は皆無です。

おサル音頭

ずっと同じ行動をし続けるプレイを世間では「サル」とか「サルプレイ」なんていいますが，不思議なことにリュウはサルプレイを得意としていません。初心者向けのキャラとはいえ，意外と頭を使います。

自分がサル化していると感じたら，それまでの自分をくずしてプレイすることをお勧めします。違った世界が見えてくることでしょう。

波動拳を撃つ間合を考えよう

スーパー頭突きを大キックでめくれ！

特別編10

見つめられると照れるでごわす E.Honda

Shirai Isao 白井 五三雄

E.本田の魅力

本田といえば頭突き，張り手，ナイスバディ，ふくらはぎ……などが思い浮かぶことでしょう。彼はストリートファイターシリーズのなかではリュウたちに次ぐ古株です。楽しい思い出。思わず涙が出てしまうちょっぴりほろ苦い思い出。歴史が長いだけに，いろいろな思いがこの本田には詰まっていることでしょう。

それでは，対戦で苦しめられた思い出しかない私が攻略を書いていきます。

基本戦術

本田はパワーがあり，飛び道具がなく，足が遅いという典型的なパワーキャラです。この手のキャラはとにかく相手に近づき相打ち覚悟で戦うのが基本です。

いかに近距離戦に持ち込むかが勝敗を決めます。まず苦手な遠中距離戦です。間合を詰めるときは歩いて詰めてください。適当に強頭突きを出すと連続技を食らいます。次に飛び道具対策です。遠中距離の間合で飛び道具を打たれたらジャンプで避けるか防御しか選択がありません。なるべく垂直ジャンプで避けるようにしたほうが無難でしょう。弱波動拳などは意外と避けるのが難しいので要練習です。

そして，なんとか中距離以内に相手を捕らえたとします。ここからが難しいので注意してください。まず安易な垂直ジャンプはしてはいけません。それよりも一歩前に出て飛び道具を防御し相手を威圧するか，立ち大キックで相打ちを狙ったほうがいいです。このじわじわと迫っていく過程が本田のすべてだと思っていいでしょう。

近距離戦になれば本田ペースです（ザンギなど一部のキャラを除く)。まずはレバーを斜め下に入れて溜めを作ります。相手が血迷って飛び道具を出そうものならスーパー百貫。フェイントで昇龍拳などの空振りを誘ったり，防御しがちな相手なら百裂張り手で削ったりするのも効果的です。もちろん飛び込んでくれば弱頭突きや小中スーパー百貫です。こうやって書くと無敵の強さがあるみたいですが，そうとも限りません。やはり彼にも，弱点や嫌な攻められ方があるのです。

嫌な攻められ方

プレッシャーを与えるつもりが逆にプレッシャーを与えられてしまい，画面の端に追い詰められ，そのままペースを握ることができずにやられる……。本田に限らずすべてのキャラにいえる負けパターンです。こうならないように戦わなければいけないのですが，もしなってしまったときのことを考えてみます。うまいリュウ，ケンに鳥籠（波動拳地獄）をやられると相手がミスするのを祈るしかありませんが，それ以外ならまだチャンスはあります。近い間合ならスーパー百貫。遠いならば垂直ジャンプなどを巧みに使い分け，相打ち覚悟で脱出を計ります。何ラウンドやっても自分が追い詰められてしまうのでしたら，おそらく相手の腕のほうが上です。修行しましょう。

この間合なら……

ソニックも安心

バルログ幸せの図

このほかに溜め系キャラの宿命として，頭突きの溜めができていないときに飛び込まれることは致命的です。でも安心してください。彼には頼もしい通常技が揃っています。主なところは近距離立ち大パンチ，しゃがみ中パンチ，しゃがみ中キックなどです。2月号のベガのところで紹介した，打点を考えた対空技を合わせて行えばかなり強力です。

まとめ

結局どう戦えばいいのか（そんな決まりなんてありませんが，どんな戦い方があるのか程度に考えてください)。飛び道具をもったキャラには積極的に前進し，ザンギやホークなどの接近戦専用キャラには多少消極的に，そのほかは前進しながら待つといった感じだと思います。なかには，「座ってばかりで何が日本男児じゃ～！　正々堂々と勝負しろ～！」という方もおられることでしょう。ごもっともです。そんなときは「立ち本田」を研究しましょう。対空は通常技のみ。頭突きやスーパー百貫は起き上がりや連続技のときのみ使用。頭突きの恐れがない分相手の動きが活発になって面白いかもしれません。類義語に「立ちガイル」などもあります。

本田ファンのあなたも，足フェチなあなたも，ほかのキャラ同様，愛してあげてくださいね。

SoundEffects

もはやコンピュータによるサウンドは単純な電子音ではなくなりました。PCM音源の普及によって，最近では自然界からサンプリングされたリアルなサウンドが当たり前のように使用されるようになってきています。シンセサイザでリアルな音を作るために苦労していたのは遠い過去の話となり，むしろ，近ごろの音源を評価していると「電子楽器音的」な音色が不足しているように感じられることすらあります。それはコンピュータミュージックが普及するための条件として，音源の「音色を固定すること」を必要としていたため，いっそう顕著になっているといえるでしょう。
そういった音源が音楽を手軽にし，誰もが楽しめるようになったのは間違いありません。しかし，音を思いっきりいじり回して音色を「創る」醍醐味がなくなり，プリセット音をただ垂れ流すことによる個性の欠如が表れ始めたのも事実です。
音源の祖である，アナログシンセサイザというものはエフェクタの塊のようなものでした。それは，多数の高周波成分を含んだ源発振音を何重にもフィルタリングすることで音色を構成するという代物です。自分が使う音は自分で作らずには，なにもできないという世界でした。
効果音は「音楽」ではなく「音」の世界です。楽器というものに縛られないぶんだけ，音楽よりは自由な展開ができます。
典型的なシンセサイザであるFM音源であれば音色はいかようにも変えることができます。しかし，その制御には小回りの利くドライバとノウハウが必要です。PCM音はどうでしょうか？　録音再生機能しか持たない音源もデータを演算／合成することであらゆる加工が可能になります。
効果音はコンピュータと人間のあいだのコミュニケーションの手段として機能します。豊かなSoundEffectsはより豊かなコミュニケーションを生み出していくのかもしれません。

CONTENTS

[特集] SoundEffects

ミュージックデバイスの基本

ボクらのベストフレンド「FM音源」

Nishikawa Zenji 西川 善司

X68000での音楽処理の中心となるFM音源
ここではFM音源の基本的な動作原理から
具体的な効果音データの作り方までを見ていこう

ゲームのVIP「ミスター効果音」

世間のゲームなどの効果音は現在もっぱらPCMが主流だ。しかしPCMデータというのはやたらメモリ資源を消費する。CD-ROMという大容量メディアが出てきたお蔭でその容量の問題は解決したんじゃないの？ という人もいるが実はそうでもない。結局メインメモリ上にそのゲームのそのシーンで使うPCMデータを読み出しておかなければ，鳴らす瞬間瞬間に使用できない。音楽CDのように読み出したPCMデータをそのままDA変換するだけの単純処理とはわけが違うのだ。パンチが敵にヒットしてからCD-ROMをアクセスしに行ってしばらくしてから「ドカ」とか鳴ったのでは臨場感もクソもあったものではない(一部にそんなソフトもあるけどね)。

CD-ROMで容量が増えても効果音の種類はたくさん収録できても，一度に使用できる数はそのシステムのメモリ容量に依存するのだ。最新ゲーム機の多くは平均500Kバイト程度の効果音用エリア(ほかの目的にも使われることがあるだろうが)を搭載しているようだが，これを手放して大きい！ と評価はできないだろう(ましてや64Kバイトしか持ってないスーパーファミコンなんて……)。500Kバイト程度だとCDレートのPCMデータ(16ビット/44.1kHz)ならば6秒程度しか収まらないのだから。

前置きが長くなった。そういう状況を踏まえて考えると，少ないメモリ資源でそこそこの音が作れるシンセサイザが再び脚光を浴びることになる。音の「リアル」さという面ではPCMに10歩も100歩も譲った感があるが，たった数十個のパラメータで任意の「音」が作れるというのは確かに魅力的だ。シンセサイザでは再現できない自然界のリアルな音をPCMで鳴らし，それ以外はシンセサイザで，というのが理想か。あ，自然界にはないシンセサイザでしか作れない音，というパターンもあるか。

究極のシンセサイザ? FM音源

「よいシンセ」の条件を，

1) 音色データに汎用性がある
2) 音色パラメータの数は多すぎない
3) 音が自由に作れる

とすると，やはりFM音源は「究極のシンセ」の称号を与えるのにふさわしい。

世の中にはさまざまなシンセサイザがあるし，そういったもののなかにはFM音源よりも魅力的な音を作り出せるものもあるが，やたら高価だとか，音色パラメータの数が多いとか，結局PCM波形をソースにしているとか，たいていいくつかの短所があったりする。上の条件のすべてを満たしているものは現在FM音源しかないのだ(反論殺到の予感)。

X68000の事情で考えてみよう。X68000はAD PCM音源が1声，そしてFM音源が8声のサウンド機能を持っているが，アクションゲームの効果音を考えた場合，なにからなにまでAD PCMで鳴らしてしまうのは少々短絡的だ。敵を連続破壊しているときにAD PCMが爆発音で手一杯なので自機のショット音が聞こえないなんて状況が頻発してくる。やはりFM音源のほうでも効果音を鳴らすようにしないと，うすっぺらな手ごたえのゲームになってしまう。

もう効果音はAD PCMだけに任せていてはいけないのだ。

FM音源の音色は

さてかなり強引に「究極のシンセ」の称号を与えられてしまったこのFM音源で効果音を作るには，やはりまず最初にその音色を作ることから始めなければならない。

するとここで「FM音源で思いどおりに音色が作れません。どうしたらよいのですか」という超メジャー級質問が浮上してくる。これに対する答えは，実は決まりきっているのだ。

どうしようもないです

これがその答えだ。いきなり怒って編集室に電話してきたりする人もいそうなのでフォローしておこう。普通の人間にはそんなことはできないという意味なのだ。どうしてか。

FM音源も結局「音」を作るシンセサイザ(合成装置)なのだから，結果的に音の波形を出力することになる。さて，ここでバイオリンの音色を作りたいとして，はたしてその波形を頭の中にイメージしたり，紙に描けるだろうか。もちろんその方面の研究者ならばある程度は可能だが，常人にはまず無理な注文だろう。

でも世の中には思いどおりにFM音源の音色を作ってしまう人がいるにはいる。彼らはどうしてそんな芸当ができるのだろうか。それは彼らが単純にFM音源のパラメータとそれからの出力音の関係を感覚的に覚えているからなのだ。もちろん彼らもFM音源に触りたての頃はなにがなんだかわからなかったに違いない。しかし何回もパラメータをいじっていくうちにその音との関係を経験的に学んでいったのだろう。だから先ほどの質問にもうひとつ答えを添えるならば「音色を作りまくればそのうち道が開ける……」ということになろうか。

FM音源の音色の仕組み

突き放したような感じもするここまでの展開だが，ちょっと退いた読者をたぐり寄せて少しFM音源の動作原理について見てみよう。もしかしたらここから先を理解できれば，いままでよりは思いどおりにFM音源の音色を作れるようになるかもしれない(すげー自信なさそうな文だな)。

音の基本的なものにサイン波(sin wave)という音があるが，これは「ぷー」という実に単純な音色だ。これを式に表すと，

$$o=\sin(\omega t)$$

(時間経過＝t,角振動数＝ω,出力結果＝o)

となる。角振動数ωはこの「ぷー」という音色のなんに相当するかというと音高(pitch)に相当する。ここを変えれば「ぷー」でドレミを演奏することができるのだ。

リスト1がサイン波のPCMデータを出力するX-BASICプログラムだ。これを実行すると 'sin.p16' という16ビットPCMデータファイルがカレントドライブに生成される。「ω」の値を変えると生成される音の高さが変わるのを確認してほしい(16ビットPCMファイルはZ-MUSIC標準ツールのZVT.Xなどで再生できる)。

FM音源の音色というのは実はこのサイン波の音が基本になっている。FM音源はこのサイン波をモジュレータ(オペレータ)で加工しその出力を別のモジュレータに入力し，これを繰り返し最終的に音声出力信号に変換している。基本式は以下のようになる。

$$o=D\times\sin(\omega t+(\text{他オペレータの出力結果})$$

Dは出力結果の倍率を決めるパラメータ，波形でいうならばすなわち振幅に相当する。

これらを踏まえたうえで，FM音源の発声の秘密を見ていこう。図1のような2オペレータ直結を例にして解説するとしよう。

まずOP.1の出力o1は，

$$o1=D1\times\sin(\omega 1t)$$

のようになる。これは単純なサイン波の出力だ。D1はOP.1の振幅を決定している。これがOP.2へ入力されるので，

$$o2=D2\times\sin(\omega 2t+o1)$$

(D2はOP.2の振幅)

がOP.2の出力結果だ。

もう勘のよい人は気づいたと思うが，DnはFM音源のパラメータでいうところのTL（トータルレベル）に相当する。

編注：TLは減衰量で示されるので，本当はOL（アウトプットレベル）と呼ぶほうが適当だが，以下，TLで代用する。

たとえばD1の値を小さくすればOP.2へ入力される値o1が小さくなるので，OP.2のsin()の「()」の中身としてはω2tの影響力が大きくなる。つまり，

$$o2=D2\times\sin(\omega 2t+o1)$$

が，

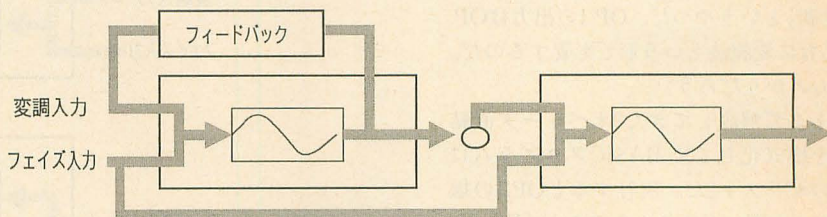
$$o2=D2\times\sin(\omega 2t)$$

に近くなるわけで，つまりサイン波の形に近くなる……すなわち柔らかい「ぽー」という音に近くなるというわけだ。

反対にD1を大きくすれば歪んだ値がOP.2に入力されるので，OP.2の波形はギザギザしたものになり結果として音も「じゃー」というようなノイジーなものになる。トータルレベルをいじると音の性質がずいぶんと変わってくるが，そのわけはこういうことだったのだ。

話を戻す。OP.2はこれまでの演算結果をそのまま音として出力するが，その前にD2によって波形の振幅を変えられる。この振幅は，そう，音量パラメータに相当する。FM音源でいえばキャリアのTLに相当するわけだ。これでどうしてキャリアのTLが音量に相当するかがわかっただろう。

そういえばωnはいったいなんのパラメータなのだろう。最初のサイン波の例だと音高に相当した。ωの値が大きければ高い音に，小さければ低い音になった。もちろんこの2オペレータ直結の例でも同様だ。しかしもうちょっと根が深い。順番に見ていこう。

図1　基本の2オペータ直列構造

リスト1　SIN.BAS

```
   5 /* サイン波
  10 screen 2,0,1,1
  90 int t,fh
 100 float C1,M1,f=0,mul=256,tt=64,w=1
 110 dim float ang(255)
 120 dim char  out(65535)
 190 /* ang:角度テーブル値配列  tt:角度テーブルの周期  mul:出力倍率
 200 for i=0 to tt-1:ang(i)=(f/(tt/2))*pi(1):f=f+1:next
 210 input "ω=",w
 250 fh=fopen("sin.p16","c")
 300 for t=0 to 32767:locate 0,1:print t
 350 C1=sin(get_rad(t*w))
 450 out(t*2)=int((C1*mul)/256):out(t*2+1)=int(C1*mul) and &HFF
 500 next
 980 fwrite(out,65536,fh)
 990 fcloseall()
 999 end
1000 func float get_rad(t;float)
1030 return(ang(int(t) mod int(tt)))
1040 endfunc
```

エンベロープと音色

エンベロープとは，シンセサイザの基本用語で一連のパラメータによってシーケンスされた出力の時間的変化のことを表す言葉だ。ところで，キャリアオペレータの出力はそのまま音量の変化になると説明した。となるとキャリアのエンベロープは音量の時間的変化となるわけだ。ではモジュレータのエンベロープというのはいったいなんなのだろうか。

これは音色の時間的変化に関係してくる。モジュレータのエンベロープというのは，モジュレータが時間的に出力割合を変化させていくというものであるから(もちろんアルゴリズムにもよるが)変調の度合を時間的に変化させるということになる。変調の度合が変われば音色の本質的な部分も変わるので結果的に音色の時間的変化となるわけだ。

で，音色を作るときに初めからこの音色の時間的変化というものを考えて作るのは非常に困難なので，これは後回しにしてもよい。

ここで，音色をゼロから作る場合の具体的な手順を述べておこう。

一般的に，音色をゼロから作っていく場合にはまず，アルゴリズムを決めることだ。これを決めるのにも随分と経験が必要だが。そこで次に，各オペレータのARは全部31，RR以外のDR，SR，SL，KS，DT1，DT2をすべてゼロにする。それでフィードバック，TL，MULをエディットしていき，音色の輪郭部分を確定していく。「だいたいこんな感じの音色でいいや」というレベルまでをこの作業で行うのだ。すでにある音色ライブラリから自分のイメージに近い音色を持ってくる場合はここまでの処理がさぼれるわけだ。

最後に音色の時間的変化，つまりエンベロープのエディットに入る。慣れてくればキャリアのエンベロープを決めるのはかなり気楽にできるようになる。しかしモジュレータのエンベロープを決定するのは熟練者になっても難しい部分だといえるだろう。

まずω1によってある周波数を持った出力がOP.1から発信される。そしてこれがOP.2へ入力される。OP.2側のsin()はこのOP.1からの入力と自分のω2を加算して，これを新たな角振動数ωとして結果を出すことになる。三角関数sinの出力を新たなラジアン角のパラメータにしているのだから数学的には次元の転換が起きてしまっている。だからこのあたりはイメージ的に非常に難解である。

かなり簡略化していってしまうとOP.1の出力した周波数がOP.2に乗り移るといった感じか。つまりシンセサイザ用語でいう「変調」というやつだ。OP.1の出力はOP.2の出力に変調波という形で影響するのだ。もうおわかりだろう。

いままで解説してきた2オペレータ直結の例を模式化してX-BASICプログラムにしたのがリスト2だ。実行するとOP.1の振幅「D1」を聞いてくる。そのあと「FB」を聞いてくる。これはFM音源のパラメータのフィードバックに相当するものだ。フィードバックとはそのオペレータの出力を次回の入力にしてしまうことをいう。OP.1は通常，

$o1 = D1 \times \sin(\omega 1t)$

なのでo1は単純なサイン波になってしまうがこれを前回のo1の値を次の計算から自分自身へ入力してやるのだ。プログラム的に記述するとこんな感じだ。

```
fb=0
loop:
    o1=D1×sin(ω1t+fb)
    fb=o1×FB
        ⋮
    goto loop
```

FBの値を大きくすれば自分の入力に戻るときに増幅されるので，以後の出力に大きく影響が出るようになる。

「FB」の値を入力すると次に「ω1」「ω2」を聞いてくる。これらに答え，出力する16ビットPCMファイルの名前を入力したらしばらく待つ。ファイルが書き出されたらZVT.Xなどで出力された16ビットPCMファイルを再生してみよう。先ほどのサイン波の音よりもかなり豊かな音色になっているはずだ。

ところで，このサンプルから4オペレータとかそれよりもっと多いオペレータ数のFM音源モドキのプログラムを作ることも簡単だ。オペレータがいくつも直結されているとすると，式は，

$on = Dn \times \sin(\omega nt + on\text{-}1)$

のような級数になる。いっそう複雑な波形となることを予感させる。

4オペレータのFM音源はアルゴリズムと呼ばれる8種類のオペレータ同士の接続の仕方を選択できるのだがこれについても簡単に解説しておく。たとえば図2のような並列構成ならば出力は，

$o1 = D1 \times \sin(\omega 1t)$

$o2 = D2 \times \sin(\omega 2t)$

$o = o1 + o2$

（oが最終出力となる）

となる。単純な波形の足し算，すなわち和音的な出力結果となる。

直列並列を組み合わせた図3のような場合は，

$o1 = D1 \times \sin(\omega 1t)$

$o2 = D2 \times \sin(\omega 2t)$

図2　2オペレータ並列の場合

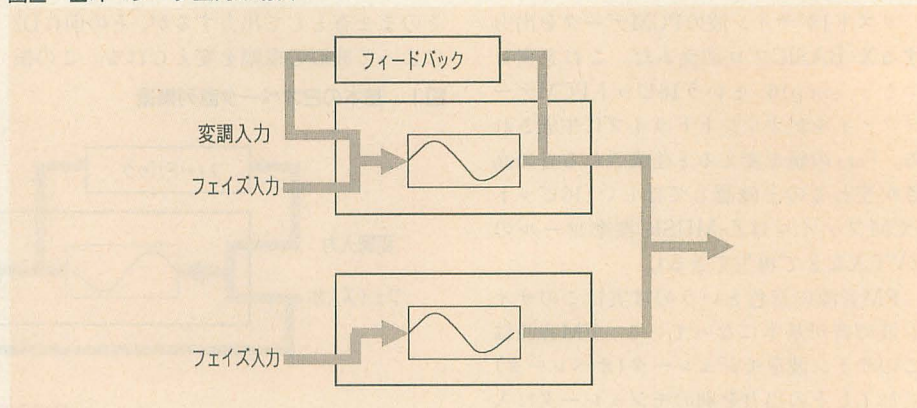
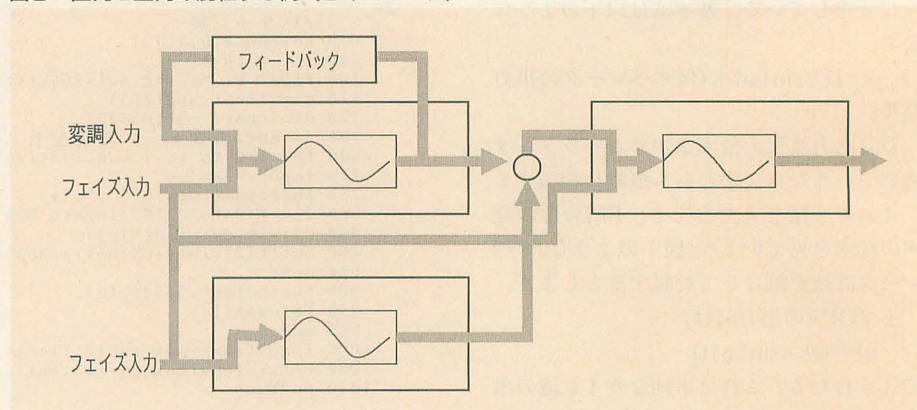



図3　直列と並列の混在する例（3オペレータ）

リスト2　2OP.BAS

```
   5 /* 2オペ直結
  10 screen 2,0,1,1
  90 int t,fh
 100 float fbsw=0,T1,C1,M1,FB=0,f=0,mul=256,tt=64,w1,w2
 110 dim float ang(255)
 120 dim char  out(65535)
 130 str fn[100]
 190 /* ang:角度テーブル値配列  tt:角度テーブルの周期  mul:出力倍率
 200 for i=0 to tt-1:ang(i)=(f/(tt/2))*pi(1):f=f+1:next
 240 input "D1(0.0～1.0～9.9)=",T1
 250 input "FB(0.0～1.0～9.9)=",fbsw
 260 input "ω1(0.0～1.0～9.9)=",w1
 270 input "ω2(0.0～1.0～9.9)=",w2
 280 input "filename:",fn
 290 fh=fopen(fn,"c")
 300 for t=0 to 32767:locate 0,6:print t
 330 M1=T1*sin(get_rad(t*w1)+FB)
 350 C1=sin(get_rad(t*w2)+M1)
 360 /*print int(C1*mul)
 450 out(t*2)=int((C1*mul)/256):out(t*2+1)=int(C1*mul) and &HFF
 490 FB=M1*fbsw
 500 next
 980 fwrite(out,65536,fh)
 990 fcloseall()
 999 end
1000 func float get_rad(t;float)
1030 return(ang(int(t) mod int(tt)))
1040 endfunc
```

$o3 = D1 \times \sin(\omega 1t + o1 + o2)$

というふうになる。なかなか複雑な出力を得られそうである。

さて，ここまでの話がわかったのだとしたら，ブラスの音を作ってみろとかいわれて「あのオペレータのマルチプルを○○にしてトータルレベルを××に……」とできるようになるには相当な経験が必要だということもわかっただろう。

トータルレベルとマルチプルはかなり音色の本質的な部分を決定する重要なパラメータであるということは覚えておいて損はない(これについてはまた後述)。しかし，これらをどう変えたからどんな音になるというのは経験的に知っていくしかない。初心者は鍛練あるのみなのだ。

それにしても4オペレータFM音源の動作をエミュレートするような音色作成ツールがあったら便利かもしれない。各オペレータの出力波形や最終出力音色の波形などがパラメータをいじるたびにリアルタイムに見られれば，いまよりもうちょっと音色が作りやすくなるかもしれない。

効果音はアイデアだ！

それでは話を効果音に戻す。

FM音源で音色を作ることはかなり難しいことがわかったわけだが，それなのに効果音なんて作れるのだろうかと心配する人もいると思う。まぁ，作るのはやさしくはないのだが決してできないというほど難しくない。フェルマーの定理を証明するよりよっぽどやさしい(そりゃそうだ)。

実はFM音源で効果音を作る場合は，まず音色を作ることを想像してしまうが，実はこれは後回しでもよい。

PCMの効果音を作る場合はネタを集めるのにいろんなCDを聞いたり，テレビやビデオを見たり，散歩したりいろいろ苦労するのだが，それは写実的にリアルなものを作ろうとするからだ。どのみちFM音源でリアルなものは作れないのだから，初めから割り切ってしまうのもひとつの手だ。もちろんPCMまっ青の超リアルサウンドを作れないとはいい切らないが，誰もが皆X68000版アフターバーナーの爆発音のようなあんなスゴイ写実的な音が作れるわけではない。

発想の転換を行うのだ。たとえば，テレビアニメのサザエさん家の猫「タマ」が走り去る音を思い出してみてくれ。ぽよよよんとかいうシンセタムのような，そこらへんのドラネコなんかじゃ絶対出さない音を発しているはずだ。同種の音に鉄腕アトムが歩く音，ドラえもんが走る音，などがある。で，テレビを見ている人はあまりそれを変だとは思わない。この音が画面にハマっているからなのだ。

実はFM音源の場合に限らず，非写実的な効果音を作る場合は「音色」と同じくらい，どんな音にしようかという発想/デザインが重要なのだ。場合によっては「音色」よりも重要であるかもしれない。

先ほどはアニメを例に挙げてしまったが，ゲームにもこういう非写実的な音はかなり多く使われている。たとえばゼビウスとかグラディウスシリーズの雑魚敵の破壊音を思い浮かべてみよう。「チャリ」とか「ピャリ」というちっちゃなガラス細工を壊してしまったような可愛い音だ。「ズガーン」とか「ボカーン」とかいうリアルな爆発音ではない。

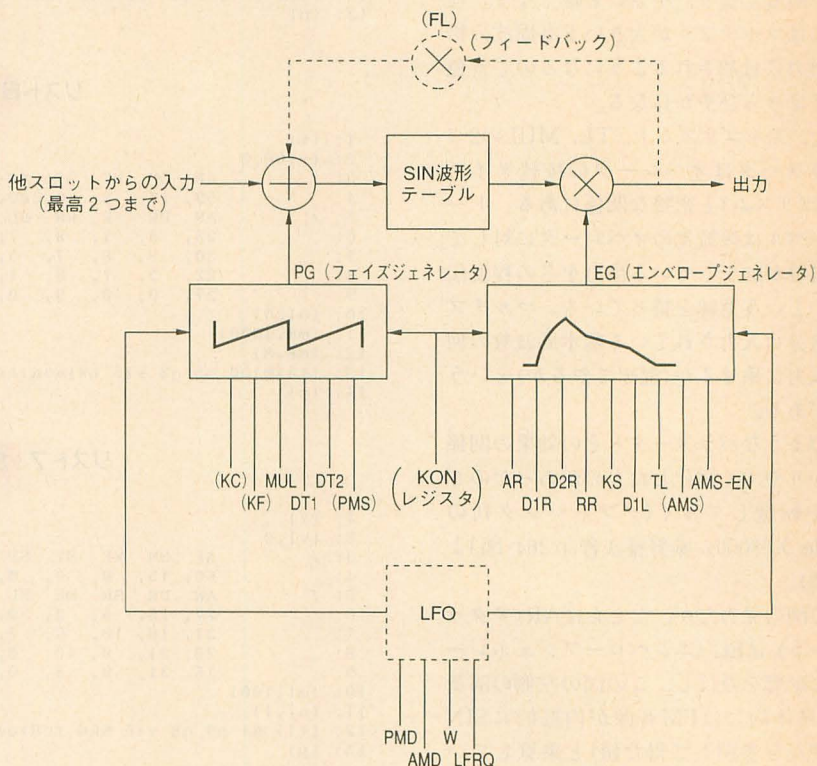
まず効果音はアイデアありき，なのだ。

音色そしてコーディング

アイデアが固まって音色を作ることになったとして，さてどーするか。もちろんいきなり音色を作り始めてもいいが，まずは自分の頭に思い描いた音にいちばんイメージの近い音色を音色ライブラリからひっぱてくる。

音色ライブラリは最初のうちは市販のでもいいが，作り出した音色は自分専用のライブラリとして構築するのがよい。自分のライブラリが充実してくると，頭に描いたイメージの音色がすぐ取り出せるようにな

図4　スロットの基本構造

リスト3　LASER1.ZMS

```
 1: (i)
 2: (v2,0                                                          /レーザー
 3: /          AF   OM   WF   SY   SP PMD AMD PMS AMS PAN
 4:            58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /          AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
 6:            12,  5,  1,  5,  5, 24,  1,  0,  0,  1,  0
 7:            16,  4,  1,  5,  2, 30,  1,  0,  3,  0,  0
 8:            29,  6,  5,  5,  2, 16,  2,  0,  3,  2,  0
 9:             9,  4, 31,  8, 10,  0,  2,  4,  3,  1,  0)
10: (m1,100)
11: (a1,1)
12: (t1) @2 c5 q8 v16 p3 @k0 g8
13: (p)
```

るだろう。

で，その取り出したる音色をエディットしていくわけだが，すでに述べたとおり，このエディットに関する適切なアドバイスというものはない……というか自分で経験的に把握していくしかないのだが，これではあまりにも不親切なので少し一般的なエディット方針を述べておく。

アルゴリズムによるが，モジュレータ(キャリアでないほうのオペレータ)のTL（トータルレベル)を0に近くすると音はノイジーになってくる。127に近くすると丸い感じになっていく。ここを127にしてしまうと次に接続しているオペレータに対して影響を及ぼさなくなってしまう。

あとはMUL（マルチプル）が倍音を作り出す重要なパラメータなのでここを中心にいじるといい。かなり大雑把な表現になってしまうが，このマルチプルの値が大きいと音が明るくなり，小さいと暗くなる。理論的にはマルチプルが大きいと高周波の倍音が出力に付加されるようになるので音色としてはきらびやかになる。

また，アルゴリズム上，TL，MULの2つのパラメータはオペレータの接続タイプ(アルゴリズム)と密接な関係にある。トータルレベルは接続先のオペレータに対して，いま自分が加工している出力をどの程度伝えるかという意味を持っている。マルチプルは自分に入力されている基本周波数の何倍を入力に乗せるか(混ぜてやるか)という意味がある。

このようなパラメータとその効果の関係をわかりやすく図化したものがあったのでこれを転載しておく(ソフトバンク刊の『Inside X68000』桒野雅彦著/p.264-図3より転載)。

この図の見方だが，たとえばAR(アタックレート）はEG（エンベロープジェネレータ）に影響を及ぼし，この図の左側の演算結果(具体的にはFM音源が内部的にSINテーブルを参照して得た値)と乗算している(具体的には出力程度を決定している)。

図中のPG（フェイズジェネレータ）とはその周波数に入力された基本周波数に相当する。たとえば高い音を発音するときには基本周波数は高い周波数が入力される。低い音ならば低い周波数になる。

実際に効果音を作ってみよう

で，最後にいくつかの効果音のサンプルを示すとしよう。

まず，リスト3～6は一般的なレーザー砲

リスト4 LASER2.ZMS

```
 1: (i)
 2: (v6,0                                                     /レーザー
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           58, 15,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           12,  5,  1,  5,  5, 24,  1,  0,  0,  1,  0
 7:           16,  4,  1,  5,  2, 32,  1,  2,  7,  0,  0
 8:           29,  6,  5,  5,  2, 16,  2,  0,  3,  2,  0
 9:            9,  4, 31,  8, 10,  0,  2,  4,  3,  1,  0)
10: (m1,100)
11: (a1,1)
12: (t1) @6 o4 q8 v16 p3 @k0 (g8>g)
13: (p)
```

リスト5 LASER3.ZMS

```
 1: (i)
 2: (v8,0                                                     /LASERの発射音
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           59, 15,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           31,  8,  1,  8,  7, 20,  2,  1,  5,  3,  0
 7:           31,  8,  8,  7,  5, 20,  1,  3,  1,  1,  0
 8:           31,  3,  7,  8,  1, 21,  1,  2,  3,  0,  0
 9:           31,  0,  0,  9,  0,  4,  2,  3,  5,  2,  0)
10: (m1,100)
11: (a1,1)
12: (t1) @8 o6 q8 v16 @k0 (c4>c)
13: (p)
```

リスト6 LASER4.ZMS

```
 1: (i)
 2: (v100,0
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           59, 15,  2,  0,200,127,127,  3,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           23,  8,  1,  8,  7, 21,  2,  1,  5,  3,  0
 7:           30,  8,  8,  7,  5, 20,  1,  2,  1,  1,  0
 8:           22,  3,  7,  8,  1, 20,  1,  1,  3,  0,  0
 9:           27,  0,  0,  9,  0,  0,  1,  2,  5,  1,  0)
10: (o120)
11: (m8,1000)
12: (a8,8)
13: (t8)@100 o5 q8 v15 L8(a>a)&v13(a>a)&v8(a>a)
14: (p)
```

リスト7 SCATTER1.ZMS

```
 1: (i)
 2: (v1,0                                                     /発射音
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           60, 15,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           27, 15,  5,  2,  0,  0,  0,  0,  3,  1,  0
 7:           31, 18, 18,  6,  7,  0,  0,  0,  3,  2,  0
 8:           22, 31,  0, 10,  0, 42,  0,  7,  7,  0,  0
 9:           15, 31,  0,  8,  0,  0,  2,  1,  7,  0,  0)
10: (m1,100)
11: (a1,1)
12: (t1) @1 o3 q8 v16 @k0 (c8>c)
13: (p)
```

リスト8 SCATTER2.ZMS

```
 1: (i)
 2: (v14,0                                                    /レーザー
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           58, 15,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           18,  8, 16,  4, 15, 22,  0,  1,  0,  0,  0
 7:           11,  8, 16,  4, 15,  0,  0,  2,  0,  0,  0
 8:           28,  8,  9,  5, 14, 35,  0,  6,  0,  0,  0
 9:           31, 16, 31, 15, 14,  0,  0,  0,  0,  0,  0)
10: (m1,100)
11: (a1,1)
12: (t1) @14 o4 q8 v15 p3 @k0 (c8>C)
13: (p)
```

の発射音のパターンだ。音色はDT2（ディチューン2）を効果的に使っているパターンが多いのに気づく。

ディチューン2とはディチューン1よりも大きめのしかも先ほど述べた基本周波数の1倍～2倍の非整数次倍音を作り出すパラメータだ。一般的な楽器の音色にこのパラメータを大きくするとアルゴリズムによってはとんちんかんな音を出してしまうことがある。しかし，効果音用の音色のような複雑な響きを持った音色を作りたい場合は大活躍するパラメータなのだ。試しにリスト3から演奏してみるといい。たった1チャンネルの音色のはずなのに単音でないような響きを感じるだろう。これが非整数次倍音の効果だ。

さて，リスト3のレーザーは純粋にレーザーの音色で発音しているだけだが，リスト4～6のレーザー音は音程の変化をつけてある。チューンといったような音はピッチをポルタメントなどで滑らかに変化させたりすると面白みのある音になる。

リスト7～8はノイズっぽい具体的な輪郭を持った音ではないのだが，これもピッチ変化を与えて効果音ぽくしている。ADPCM音をいまのようにドラムなどで使用できなかった時代に，このようにノイジーな音のピッチを丸めてスネアドラムなどの代わりにしていたものだ。

リスト9は秘密基地とかのゲートを開けるような音だ。音色はほとんどレーザーと変わらないが，ピッチ変化を効果的に使って雰囲気を出している。まず短時間のピッチアップそしてこの音程を持続させながらビブラートをかけている。このあたりの周期的な揺らぎがなにか重いものを転がしているような感じを出している。そして最後に，短期間のピッチダウン。これでゲートが開ききった，ゲートを開ける駆動装置のスイッチの停止音みたいなニュアンスを出している。

リスト10～14はリリース音を効果的に使ったものだ。リリース音とは消音処理したあとに残る余韻の音のことである。FM音源ではRR（リリースレイト）と呼ばれるパラメータでこの余韻時間を決める。このパラメータはFM音源で音楽などを作っている場合には効果が出にくい。というのもFM音源で音楽を作っている場合は，リリース音が鳴る暇もなく次の音を発音しているからだ。リスト10からの効果音はこのリリースをあえて生かしてみたものだ。

リスト10～13は衝突したときの音やなにかを斬ったときの音なのだ。いままで述べ

リスト9 GATEOPEN.ZMS

```
 1: (i)
 2: (v12,0                                                        /ゲートオープン
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           59, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           23,  8,  1,  8,  7, 31,  2,  1,  5,  3,  0
 7:           30,  8,  8,  7,  5, 20,  1,  2,  1,  2,  0
 8:           22,  3,  7,  8,  1, 20,  1,  1,  3,  0,  0
 9:           27,  0,  0,  9,  0,  0,  1,  2,  5,  1,  0)
10: (m8,1000)
11: (a8,8)
12: (t8)@12 o3 q8 v14 p3 @m60@h24@s3(a16<a)&a4.&(a16>a)
13: (p)
```

リスト10 CHOWEEN.ZMS

```
 1: (i)
 2: (v4,0                                                         /チョウィーン
 3: /         Af  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           31,  5,  3,  0,  0, 14,  1,  0,  3,  3,  0
 7:           28,  0,  0,  5,  3,  0,  0,  1,  0,  1,  0
 8:           31,  0,  0,  5,  2,  0,  0,  2,  7,  0,  0
 9:           28,  0,  0,  5,  2,  0,  0,  3,  7,  2,  0)
10: (m1,3000)
11: (a1,1)
12: (t1)@4 o7 q8 v12 p3 @k0 L32(c>c)&(c<e)r1
13: (p)
```

リスト11 ZUGOAN.ZMS

```
 1: (i)
 2: (v4,0                                                         /ズゴーン
 3: /         Af  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           31,  5,  3,  0,  0, 14,  1,  0,  3,  3,  0
 7:           28,  0,  0,  5,  3,  0,  0,  1,  0,  1,  0
 8:           31,  0,  0,  5,  2,  0,  0,  2,  7,  0,  0
 9:           28,  0,  0,  5,  2,  0,  0,  3,  7,  2,  0)
10: (m1,3000)
11: (a1,1)
12: (t1)@4 o2 q8 v12 p3 @k0 L32(g>g)&(g<b)r1
13: (p)
```

リスト12 JINGONG.ZMS

```
 1: (i)
 2: (v3,0                                                         /チコーン
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           31,  5,  3,  3,  1, 18,  0,  0,  3,  0,  0
 7:           18,  0,  0,  5,  3,  0,  0,  1,  0,  2,  0
 8:           31,  0,  0,  5,  2,  0,  0,  2,  7,  3,  0
 9:           18,  0,  0,  5,  2,  0,  0,  3,  7,  1,  0)
10: (m1,3000)
11: (a1,1)
12: (t1)@3 o6 q8 v12 p3 @k0 L32(c<e)&(e>c)r1
13: (p)
```

リスト13 DBON.ZMS

```
 1: (i)
 2: (v15,0                                                        /ゴボーン
 3: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:           61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:           31, 11,  3,  5,  1, 17,  0,  1,  3,  0,  0
 7:           31, 11,  7,  5,  3,  0,  0,  2,  0,  1,  0
 8:           31, 11,  7,  5,  3,  0,  0,  1,  7,  3,  0
 9:           31, 11,  7,  5,  3,  0,  2,  3,  7,  1,  0)
10: (m1,3000)
11: (a1,1)
12: (t1)@15 o3 q8 v16 p3 @k0 L32(c>g)&(g<c)r1
13: (p)
```

たピッチ変化テク効果を使っているのはもはや当然で，それ以外にちょっとした工夫がしてある。

演奏データの音符部分を見てみよう。休符「r1」が見えるだろう。これが余韻部分だ。ほんとにちょっとした工夫で申し訳ないがこの休符の間，前に鳴っていた音の余韻が持続するのだ。そう休符は無音状態ではないのだ。

で，逆にこのリリース音をまったくなくしたパターンがリスト14。なにかが弾けたような音だ。

演奏データ部分に「p0」というのが見えたかな。pコマンドはパンポット(音場)を設定するコマンドだった。FM音源ではp0は音をどこにも出さないという設定なので，これを音符が消音したあとに使用すれば，リリース音もなく，ピタッと無音になるのだ。音楽を作る場合にもブレイクフレーズでこれを効果的に使うと面白いかもしれない。

で，このリスト14のパターンはなんか意味ありげに音符が記述してあるが，これは半分適当で半分マジでシーケンスしたもの。まず，出したい効果音の音を頭で思い描き次に口で真似してみる。そしてこれを採譜してみてこんな感じかなと音符に起こす。いや，こうじゃない，こうだろうと何度もいじって自分の好みのものにしていく。こうしてできたものなのだ。口で効果音をいってみるのは意外と重要。恥ずかしくてもやるように。

リスト15もリスト14のように苦心したパターンだ。コミカルなキャラクターがポテっとこける音。漫画チックな音だ。音符を見る限り単なるオクターブ上昇するだけのものだが，なにに苦労したかというと「音色」だ。頭の中で描いた音のイメージから音符はこれだというのは想像できたのだが，これで「ポテ」という音に鳴ってくれる音色を作るのが大変だったのだ。こういうパターンもあるのだ。

リスト16～18は，別の効果音のパターン。日常鳴っている音で「小刻みに鳴っている音」「連続で鳴っている音」というものをよく耳にするだろう。このパターンだ。

リスト16はオーソドックスな電話のベルだ。「リリリリリ」というのをベルっぽい音で小刻みに鳴らしている。

リスト17は音を鳴らす時間間隔をだんだんと変化させていくもの。音のイメージとしては「戦闘機のスコープが敵をとらえ，これをロックオンする」というものなのだが，間隔をもうちょっとあけてゆったり演

リスト14 MYUWA.ZMS

```
 1: (i)
 2: (v7,0                                                          /弾ける音
 3: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:            58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:            25,  5,  3,  3,  3, 26,  0,  0,  3,  2,  5
 7:            13, 31,  0,  5,  5, 27,  0, 11,  2,  2,  5
 8:            25,  5, 11,  7,  7, 10,  0, 10,  7,  3,  5
 9:            31, 31,  0, 15,  0,  0,  0,  1,  0,  0,  0)
10: (a1,1)(m1,1000)
11: (t1)t120@707v15p3@k0@l1e<g>>cd<<e>(f8<f)p0
12: (p)
```

リスト15 POTE.ZMS

```
 1: (v1,0
 2: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 3:            56, 15,  2,  1,100,127,  0,  4,  0,  3,  0
 4: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 5:            20, 10,  0,  5,  0, 41,  0,  0,  0,  0,  0
 6:            20, 16,  0, 10, 15, 17,  0,  0,  0,  0,  0
 7:            20, 10,  0,  6, 15, 19,  0,  1,  0,  0,  0
 8:            31, 18,  0, 10, 15,  0,  1, 13,  0,  0,  0)
 9: (i)
10: (m1,100)
11: (a1,1)
12: (t1)t180 @1 o2 q8 v13 L8e<e>
13: (p)
```

リスト16 TELEPHONE.ZMS

```
 1: (v70,0
 2: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 3:            60, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 4: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 5:            31, 19, 11,  5, 10, 17,  0,  8,  3,  0,  3
 6:            25, 19, 11,  5, 10,  7,  0,  6,  2,  2,  5
 7:            19, 11, 19,  7, 10, 27,  0,  8,  7,  1, 13
 8:            25, 11, 19,  6, 10,  0,  0,  8,  7,  0, 15)
 9: (i)
10: (m1,100)
11: (a1,1)
12: (t1)@70O4V15l48|:12a:|r32|:12a:|
13: (p)
```

リスト17 PIPIPI.ZMS

```
 1: (v70,0
 2: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 3:            52, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 4: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 5:            30, 19, 30,  8, 13, 13,  0,  7,  3,  2,  0
 6:            31, 14, 28, 11,  1,  0,  0,  6,  3,  0,  0
 7:            31, 19, 13, 12, 14, 12,  3,  5,  7,  3,  0
 8:            31, 15, 10,  9,  1,  0,  3,  5,  7,  0,  0)
 9: (i)
10: (m8,1000)
11: (a8,8)
12: (o120)
13: (t8)@70 o4 q8 v15 L24aaaaL32aaaaL64aaaa@L1aaaa
14: (p)
```

リスト18 BOOON.ZMS

```
 1: (i)
 2: (v9,0                                                          /要塞の中心部
 3: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:             0, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:            16,  5,  0, 15,  3, 16,  0,  0,  7,  0,  0
 7:            16,  8,  0, 15,  2, 25,  0,  0,  7,  0,  0
 8:            18,  8,  0, 15,  2, 26,  0,  0,  7,  0,  0
 9:            16,  4,  0, 15,  3,  0,  0,  0,  3,  0,  0)
10: (m1,3000)
11: (a1,1)
12: (t1)@9 o1 v16 @k0 q8 |:20c1:|
13: (p)
```

奏すれば「硬いテーブルの上にピンポン玉を落とした」音にもなりそうだ。

リスト18は逆に時間間隔を目一杯あけたもの。「要塞の中心部のエネルギー発声装置の音」とか「生命維持装置の中で静かに脈打つエイリアン」とかそんな感じだ。低い重い音が緊張感を醸し出す。オクターブを変えるとまた違った感じになるだろう。

リスト19～20は1チャンネルエコーというテクニックだ。リストを実行して鳴る音は単なるクリック音だが，ヘッドホンで聞くと1声しか鳴っていないのになんかエコーがかかっている感じがするはずだ。これはまず鳴らしたい音を通常の音量で鳴らし，このあと音量を半分以下にして同じ音を鳴らす。以後も徐々に小さい音を連続で小刻みに鳴らす。これでこの感じが出る。

リスト21～23は音色にこだわりまくった場合だ。

リスト21はオペレータ4のアタックレートが極端に小さい。そしてこの音でピッチダウン(高い音から低い音へのポルタメント)を実行し，さらに休符「r1」で余韻を醸し出している。これは遠くから飛行機/宇宙船が飛来してくるというイメージ。

アタックレートを小さくしたのは音の立ち上がりを遅くすることによって遠くからやってくることを表すためだ。そしてピッチダウンは，ほら，車が自分の前を通りすぎるとヒューンて感じに聞こえるでしょ，ドップラー効果。あれだ。今回掲載したリストはかなり遠くからやってくる感じがするが，これは音色や演奏データをいじるなりして調整できる。

リスト22はイマイチという声もあるのだが一応車のエンジン音/走行音のつもり。リストではピッチアップ(低い音から高い音へのポルタメント)を実行してどんどん加速していく感じを出している(つもり)。洗濯機の脱水機の音みたいという意見もあったが，まーなにかが回っているというイメージは確かに表現できたようだ(?)。

最後のリスト23はハードLFOを使ったヘリコプターの音。AM(アンプリチュードモジュレーション)で小刻みに回っている感じを，PM(ピッチモジュレーション)でブレードが空気を切り裂いたときに発生する共鳴音のような感じを出している。PM/AMどちらかをオフにすればまた変わった感じにもなるだろう。

*　　*　　*

今回はなんだか一風変わったFM音源講座になってしまったが，ま，たまにはこういうのもいいんではないか，と。

リスト19　PIPI.ZMS

```
 1: (i)
 2: (v10,0                                                 /クリック音
 3: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:            52, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:            30, 19, 30,  8, 13, 13,  0,  7,  3,  2,  0
 7:            31, 14, 28, 11,  1,  0,  0,  6,  3,  0,  0
 8:            31, 19, 13, 12, 14, 12,  3,  5,  7,  3,  0
 9:            31, 15, 10,  9,  1,  0,  3,  5,  7,  0,  0)
10: (m6,1000)
11: (a6,6)
12: (t6)@10 o4 q8 v15 @k0 L32av2av1av0a
13: (p)
```

リスト20　PIPI2.ZMS

```
 1: (i)
 2: (v13,0                                                 /クリック音
 3: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:            56, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:            20, 10,  0,  5,  0, 41,  0,  0,  0,  0,  0
 7:            20, 16,  0, 10, 15, 17,  0,  0,  0,  0,  0
 8:            20, 10,  0,  6, 15, 19,  0,  1,  0,  0,  0
 9:            31, 18,  0, 10, 15,  0,  1, 13,  0,  0,  0)
10: (m6,1000)
11: (a6,6)
12: (t6)@13 o3 q8 v15 @k0 L32av3av2av0a
13: (p)
```

リスト21　JET COMMING.ZMS

```
 1: (i)
 2: (v11,0                                                 /飛来音
 3: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:            58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 5: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:            31,  0,  0,  0,  0, 17,  0,  2,  0,  2,  0
 7:            31,  0,  0,  0,  0, 10,  0,  1,  0,  1,  0
 8:            31,  0,  0,  0,  0, 29,  0,  2,  0,  2,  0
 9:             2,  0,  0,  4,  0,  0,  0,  0,  0,  0,  1)
10: (m7,3000)
11: (a7,7)
12: (t7)@11 o6 q8 v16 @k0 (e*540>b) r1
13: (p)
```

リスト22　CAR.ZMS

```
 1: (v1,0
 2: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 3:            58, 15,  2,  0,  0,  0,  0,  0,  0,  3,  0
 4: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 5:            31,  0,  0,  0,  0, 28,  0,  0,  0,  0,  0
 6:            31,  0,  0,  0,  0,  8,  0,  0,  0,  1,  0
 7:            31,  0,  0,  0,  0, 46,  0,  1,  0,  0,  0
 8:            21,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0)
 9: (i)
10: (m1,9999)
11: (a1,1)
12: (t1)@1 o2 v15 q8 (c<c)576&(c<c)576&(c<c)576
13: (p)
```

リスト23　HERI.ZMS

```
 1: (i)
 2: (v11,0
 3: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 4:            58, 15,  2,  1,220,120, 30,  7,  2,  3,  0
 5: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 6:            31,  0,  0,  0,  0, 17,  0,  2,  0,  2,  0
 7:            31,  0,  0,  0,  0, 10,  0,  1,  0,  1,  0
 8:            31,  0,  0,  0,  0, 29,  0,  1,  0,  2,  0
 9:            12,  0,  0,  4,  0,  0,  0,  0,  0,  0,  1)
10: (m7,3000)
11: (a7,7)
12: (t7)@11 o4 q8 v15 c1
13: (p)
```

[特集] SoundEffects

Z-MUSICでのテクニック

FM音源効果音のすすめ

Horie Kohtarou 堀江 孝太郎

X68000での多彩な効果音作りには欠かせないFM音源
FM音源効果音の弱点を補うための新システムコールについても解説する
ゲームプログラマはサンプルプログラムを参照してみてほしい

Z-MUSICでゲームを作る人へ

Z-MUSICは，内蔵音源はもちろんMIDIも使えるし，ライセンスはフリーだし，効果音（以下SE）処理はサポートされていたりと，ゲームに使用されることも前提に作られたミュージックドライバだ。Oh!Xで発表されたゲーム「SION2」，「PUSH BON!」，市販ゲームだったら「餓狼伝説スペシャル」，そして同人ソフトなど多くのゲームソフトで使用されている。ここでは，Z-MUSICを使った処理について説明したいと思う。なお，この解説を理解するにはOh!X BOOK「Z-MUSICシステム ver. 2.0」のマニュアルは絶対に必要だ。

ゲームにとってBGMはもちろん重要だけど，SEというのも凄く重要だ。同人ソフトなんかでSEがおろそかになってるものをよく見かけるけど，これは凄く損してると思う。レースゲームでエンジン音がないとか，格闘ゲームで相手を殴ったり，地面に叩きつけたのにスカスカした音がするとか，シューティングゲームで自機が弾を発射してるのにショット音がしないとかだと，なんかプレイしてていまひとつ盛り上がらない。

やっぱり最低限でも，レースゲームはエンジン音が速度によって変わって敵車が近くにいるなら敵車のエンジン音も聞こえてほしい。格闘ゲームだったら殴ったら「バキッ！」と痛そうな音がして，やったほうは「おりゃー」とか叫んで，やられたほうは「うっ！」とか痛がってほしい。シューティングゲームだったら「ダダダダッ」とか「ズキューン」とかショット音がして敵に当たれば「バリバリバリッ！」とダメージ音がして「ドガーーーン」と爆発してほしいな。

さてさてX68000はご存じのように内蔵音源にFM音源とAD PCMが装備されている。基本的にFM音源8音，AD PCM1音の同時発声だ。X68000でゲームを作るのであれば，このスペックでBGMとSEを処理しなくてはならない。

BGMは私はFM音源だけのBGMも結構好きなんだが，世間ではBGMにFM音源とAD PCMを使うのが普通になっている。そうするとSEはどうしよう……ということになる。

格闘ゲームならば攻撃ヒット音とか声とかが必要になり，これらはAD PCMでなければ不可能だ。こうなると，BGMのAD PCMパートは多少息切れしてもしかたがない。

Z-MUSICでAD PCMのSEを発声するときには，

　ファンクションコール$13 se_adpcm1
　ファンクションコール$14 se_adpcm2

のどちらかを使用すればいいだけだ。再生周波数は5つ，パンは3つから選べる。おまけに前にリクエストされたSEといまリクエストしたいSEとの優先レベルも指定できる。

シューティングゲームではどうだろう。自機ショットSE，攻撃ヒットSE，爆発SE，アイテム獲得SE……と，格闘ゲームよりフィーチャーが多い。これを全部AD PCMでやったらBGMのAD PCMパートは息切れじゃすまない。PCM8などのポリフォニックAD PCMドライバを使ってもいいけど，それにCPUパワーが持ってかれるのはちょっと問題だ。やはり，ここはFM音源にもSEを頑張ってもらうしかない。幸い，Z-MUSICにはFM音源の各種パラメータに作用する特殊コマンドが揃っているので効果音作りにはこと欠かないだろう。

Z-MUSICではFM音源を使ったSEを使用することも考慮して作成されている。ここで使うFM音源の効果音というのは「MMLで記述された小さな曲」というふうにとらえることもできる。音色はBGMと共用しなければならないが，数トラックにわたる曲でも大丈夫だ。

たとえば，OPMの8チャンネルすべてを使用した音楽が鳴っていたとき，別の曲(SE)が割り込んで演奏されるとしよう。するとSEで使用しているチャンネル部分が効果音に差し替わり，SEで使用していないチャンネルはなにごともなかったように演奏を続ける。SEの演奏が終了すると元のチャンネルの演奏が継続するように辻褄をあわせる……といった処理をシステムがすべてやってくれるのだ。ちょっと見ると，なんてことなさそうな気もするが，BGM中断中の音色切り換えやLFOなどの処理も進行したところからちゃんと再現するためには，SE発声中もBGM部分の演奏処理を並行して行う必要がある。結果的に，現在のZ-MUSICは64トラック分の演奏データと64トラック分の効果音データを並列処理する能力を持っていることになるのだ。

さて，Z-MUSICでFM音源のSEを発声するには，

　ファンクションコール$12 se_play

を使用する。しかし，AD PCMの場合はPCMデータを受け渡すというシンプルさだったが，FM音源の場合はZMDデータを受け渡すので結構複雑で，高速処理のためにいくつかの制約がある。なかでもいちばん面倒臭い制約はタイマがひとつしか使えないことだと思う。この制約によってSEのテンポにあわせてBGMのテンポと絶対音長を調節するという作業をしなくてはならないのだ。

私は以前，Z-MUSICを使ってSEはすべてFM音源で行うゲームをプログラムしたことがあった。BGMはMIDI対応で，音楽屋さんがMIDIデータをPC-9801のレコンポーザで作ってZMDにコンバートしたものなので，絶対音長の調整はほとんど不可能だった。

また，頻繁にSEが鳴るようならば，FM

音源でもSEが割り込むチャンネルは息切れをすることになる。FM音源は8チャンネルだから何チャンネルかをSE専用に割いてやってもいいように思う。コナミの「出たな!!ツインビー」では2チャンネル，「パロディウスだ！」では3チャンネルもSE専用に割り当てているようだ。

このように，

・BGMとSEは同タイマで処理
・SEはBGMの任意のチャンネルに割り込んで処理

という従来の仕様とまったく正反対の，

・BGMとSEは別タイマで処理
・SE専用にチャンネルを割く

の2つの条件でSE処理するためのファンクションコール$58（以下コール$58）がZ-MUSIC ver2.03から新設された。ver.2.03は最近では1994年10月号の付録ディスクに収録されている。

新設ファンクションコール$58

コール$58は外部のアプリケーションからZMDの演奏ルーチンを直接サブルーチンコールできるものだ。ゲームプログラムで用意したSEデータ（ZMD）をBGM演奏とは無関係に演奏できる。別の割り込み源を発生させ，その割り込みルーチンでこのファンクションを用いればZMUSIC.Xの音楽演奏のテンポとはまったく無関係のSE処理が行える。

引数はアドレスレジスタa5.lで，a5.lは外部アプリケーション側で用意した256バイトで構成されるトラックワークだ。ワークの内容はマニュアルやZ-MUSICソースのLABEL.S（ラベル定義ファイル）を見てほしい。またトラックワークの初期化は外部アプリケーション側で行う必要がある。コール$58の実行サンプルプログラムを作ってみたので順に解説する。

プログラムの実行

プログラムはハイスピードアセンブラHAS.X用に書かれている。アセンブルにはIOCSCALL.MAC，DOSCALL.MAC，Z-MUSIC.XソースのLABEL.Sが必要となっている。

プログラム実行は，

1)　Z-MUSICはMオプションをつけて常駐する。

　　ZMUSIC -M

2)　音色リストneiro.zmsをZP.Rで登録する。

　　ZP neiro.zms

3)　適当な手持ちの曲データをZP.Rで演奏する。

　　ZP 曲データ.zms

4)　サンプルプログラムを実行する。

　　se_tst.x

という手順だ。

プログラムでは垂直同期割り込み（以下V-SYNC)を使用している。ZP.Rのデバッグモードなどですでに V割り込みを使用していると実行できない。

キー操作は以下のとおりだ。

[+]，[－]
　SEの種類を変更する。
[8]，[2]（テンキー）
　音量を上下させる。
[4]，[5]，[6]（テンキー）
　パンを左，中央，右と変更する。
[ENTER]
　SEを発声する。
[F10]
　プログラムを終了する。

プログラム解説

SE演奏にV-SYNCを使用している。この割り込みはゲームでは必ず使用すると思うので，ほかの処理と一緒にフックしてやればいい。

プログラムではFM音源の7，8チャンネルをSE専用に割り当てているため，曲データ側でこのチャンネルを使っている場合は強制的にマスクしている。ルーチンfm78maskがその処理だ。ファンクションコール$44でもマスクできそうだができない。今回のサンプルSEはSE_0～4が1チャンネル，SE_5のみ2チャンネルのSEデータとなっている。

ルーチンv_intがV-SYNCルーチンだ。ここでは入ってすぐ自分の割り込みを，マスクとSRレジスタを操作することによってV-SYNCの上にOPM割り込みが乗れるようにしてやる。コメントにTIMER_A INT_OFFとあるがIOCS_VDISPSTで発生する割り込みは本当はV-SYNC(GPIP4)ではなくてTimerA割り込みだからだ（ややこしい）。

ルーチンint_se_playがコール$58をコールするルーチンなのだが，ひとつ注意することがある。多重割り込みによってOPM割り込みの上にV-SYNCが乗ってしまうことがあるのだ。このときZMUSIC.Xの演奏ルーチンをリエントリしてしまいSEがBGMの演奏に影響を及ぼしてしまうことがある。これを防ぐために，割り込みマスクレジスタBを見てV-SYNCがOPM割り込みの上に乗ってしまったようであればコールを見送るようにしてやる。これによってSEは厳密にいえばテンポずれしてしたような感じになるのだが，そうそう起こる現象ではないのでほとんど気にならないと思う。

付加機能としてパンとボリュームの変更ができるようにした。アルゴリズムはSE演奏前にZMDコードを直接書き換えてしまうことで実現している。この機能のためにはZMSデータは「@n Vn Pn」，すなわちZMDコードは「$A0 nn $B6 nn $B3」で始まってる必要がある。ボリューム変更は4バイト目を$80－ボリューム値，パン変更は5バイト目を左出力は$B1，右出力は$B2，出力中央は$B3に変更してやる。

ルーチンzmse_seq_wk_initがアプリケーションで用意した演奏トラックワークの初期化ルーチンだ。ほかにもプログラム実行の最初にルーチンint_se_play_initで演奏絶対チャンネルの設定をしている。

SEデータについて

今回のプログラムでは6種類のSEを収録してある。V-SYNCによって得られるSEの音楽テンポ値は，

　60(クロック)×60(秒)÷48(4分音符の絶対音長)＝75

となり，SEデータはt75で作ってやればよい。もっと速いテンポでSEが作りたいのであればTimerDを使ってやればいいだろう。私はt75でもなんとかなると思っている。

SE用のデータには以下のMMLコマンドは使用できない。

　テンポコマンド
　同期コマンド
　和音コマンド

プログラムに埋め込まれているSEデータはZMDデータからヘッダ情報（先頭10バイト）をカットしたものである。SE_0については効果音のMMLデータをプログラムソース上にコメントをつけておいた。その他の効果音ZMSファイル（se_0.zms～se5.zms）とあわせていろいろと確認してほしい。

さらなる拡張

このプログラムでは無条件に次々とSEの発声が行われるので必要がなかったのだが，ゲームではSEごとの優先レベルが存在

する。敵の爆発音が発声しているときには自機のショット音は発声されないというふうにしなければならない。

優先レベルが高いSEが発声されているときに低いSEを無視するには，優先度の高いSEが発声し終わったかをトラックワークp_not_emptyで使用チャンネル分を真面目に検査してもいいが，あらかじめ各SEの発声時間を計測しておき，カウンタを設けて発声時間カウンタが終了したら，次のSEを許可するという方法もいいかもしれない。このほうがSEのわずかの余韻ならばカットしてしまうこともできていいからだ。これができるようであればAD PCMのSEもファンクションコールの優先レベルを使わずに自分で管理するといいだろう。

最後に

実はこのSE処理はほぼ1年くらい前に作ったものだ。いま考えるとちょっと回りくどく泥くさい処理をしてる気もする。もう少し研究すれば，ひょっとしたらSE専用にチャンネルを割かずBGMとチャンネルを兼用できるようにもなるかもしれない。こうなると，8チャンネル使ったSE（というか曲だな）も鳴らせたりしていいんだけど。

やっぱりZ-MUSICでも最初からSEにタイマを2つ持ってもらいたかったな。あと音色のバンクもBGMとは別でね。SEがゲームでの用途だけでなく，アプリケーションすべてで使えるという感じにしたいからだ。たとえば，プレイヤー上でプレイボタンを押すと「ピッ」とクリック音がしたり，OS上でのエラー音をSEでやってみるのも面白そうでいいなと思っているんだけど（素直にAD PCMでやれっていわれそうだけど）。

リスト1 NEIRO.ZMS

```
(i)

.fm_vset200={
/*  AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
    31.  0.  0. 15.  0. 32.  0.  8.  0.  0.  0.
    31.  0.  0. 15.  0.  0.  0.  2.  0.  0.  0.
    31.  0.  0. 15.  0.  0.  0.  2.  0.  0.  0.
    31.  0.  0. 15.  0.  0.  0.  2.  0.  0.  0.
/*  CON FL  OP
     6.  7. 15}

(p)
```

リスト2 効果音のソース

```
==================  SE_0.zms  ========
 1: (i)
 2: (m 8.1024)(aFM8.8)
 3:
 4: (t8)
 5: @200 v16 p3
 6: /t75
 7: o4L4q8@k0@m@o
 8: |:2 @L1o2fedc>bagf_10:|
 9:
10: (p)

==================  SE_1.zms  ========
 1: (i)
 2: (m 8.1024)(aFM8.8)
 3:
 4: (t8)
 5: @200 v16 p3
 6: /t75
 7: o4L4q8@k0@m@o
 8: _8@L1|:12 o4c>f:|
 9:
10: (p)

==================  SE_2.zms  ========
 1: (i)
 2: (m 8.1024)(aFM8.8)
 3:
 4: (t8)
 5: @200 v16 p3
 6: /t75
 7: o4L4q8@k0@m@o
 8: o6a16.
 9:
10: (p)

==================  SE_3.zms  ========
 1: (i)
 2: (m 8.1024)(aFM8.8)
 3:
 4: (t8)
 5: @200 v16 p3
 6: /t75
 7: o4L4q8@k0@m@o
 8: |:5 o5g*1>>g*1<<g*1>g*1_5:|
 9:
10: (p)
11:

==================  SE_4.zms  ========
 1: (i)
 2: (m 8.1024)(aFM8.8)
 3:
 4: (t8)
 5: @200 v16 p3
 6: /t75
 7: o4L4q8@k0@m@o
 8: @L1o6cd+cfcg+
 9:
10: (p)
11:
12:

==================  SE_5.zms  ========
 1: (i)
 2: (m 7.1024)(aFM7.7)
 3: (m 8.1024)(aFM8.8)
 4:
 5: (t7)
 6: @200 v16 p3
 7: /t75
 8: o2L4q8@k0@m@o
 9: |:2 (o1c8.o3c):|
10:
11: (t8)
12: @200 v16 p3
13: o2L4q8@k0@m@o
14: |:2 (o2c8.o4c):|
15:
16: (p)
```

リスト3 se tst.s

```
 1: *          Func$58 Sample Program for ZMUSIC ver.2.03
 2: *                                        Programmed by あほみ
 3: *                                        in 1995/01/22～
 4:
 5:                  INCLUDE IOCSCALL.MAC
 6:                  INCLUDE DOSCALL.MAC
 7:                  INCLUDE LABEL.s
 8:
 9:                  .nlist
10:
11: HM_SE:           equ     6                 * SEの数
12:
13: ZM_VER:          equ     $2003
14:
15: ZM_VERs:         MACRO
16:                  dc.b    '2.03'
17:                  ENDM
18:
19: *zmaccall.mac
20: Z_MUSIC:         macro   callname
21:                  moveq   #callname,d1
22:                  trap    #3
23:                  endm
24: *
25: OPMSET:          MACRO   reg,data
26:                  bsr     opm_busy_chk
27:                  move.b  reg,$e90001
28:                  bsr     opm_busy_chk
29:                  move.b  data,$e90003
30:                  ENDM
31: *
32: INT_MACHI:       MACRO
33:                  LOCAL   V01
34: V01:             tst.w   v_cnt(a6)
35:                  beq     V01
36:                  clr.w   v_cnt(a6)
37:                  ENDM
38: *
39: AKEYSNS:         MACRO   n,m
40:                  btst.b  #m,$800+n.w
41:                  ENDM
42: *
43: B_LOCA:          MACRO   x,y
44:                  moveq   #x,d1
45:                  moveq   #y,d2
46:                  IOCS    _B_LOCATE
47:                  ENDM
48:
49: B_PUTC:          MACRO   n
50:                  move.b  n,d1
51:                  IOCS    _B_PUTC
52:                  ENDM
53:
54: DISPLAY:         MACRO   ptr,attr,x,y,len  * メッセージ表示のマクロ
55:                  lea     ptr,a1
56:                  moveq   #attr,d1
57:                  moveq   #x,d2
58:                  moveq   #y,d3
59:                  moveq   #len-1,d4
60:                  IOCS    _B_PUTMES
61:                  ENDM
62:
63:                  .list
64:
```

▶SEGA SATURNの売りの「バーチャファイター」だが，「バーチャファイター2」を一度でもプレイするとやはりアーケードがいちばん！　と思ってしまう。
岡部　祥明(20)茨城県

```
 65:                .text
 66:                .even
 67: start:
 68:          lea     work_top(pc),a6
 69:
 70:          lea     stack_buf(pc),a0
 71:          move.l  sp,(a0)
 72:          move.l  a0,sp
 73:
 74:          suba.l  a1,a1
 75:          IOCS    _B_SUPER
 76:          move.l  d0,to_user_sp(a6)
 77:
 78:          bsr     zmsc_inst_chk
 79:          bsr     int_se_play_init
 80:          bsr     v_int_set
 81:
 82:          bsr     fm78_maskon
 83:
 84:          moveq   #$10,d1                         * 768×512モード
 85:          IOCS    _CRTMOD
 86:          move.w  #18,-(sp)                       * カーソル非表示
 87:          DOS     _CONCTRL
 88:          addq.w  #2,sp
 89: *
 90:          DISPLAY mes_SNO(pc),3,0,0,11
 91:          DISPLAY mes_VOL(pc),3,0,1,11
 92:          DISPLAY mes_PAN(pc),3,0,2,14
 93: *
 94:          move.w  #$00B3,se_pan(a6)
 95: main_lp:
 96:          AKEYSNS $d,4                            * [F10]
 97:          bne     fin
 98:          INT_MACHI
 99:          bsr     _put_seno
100:          bsr     _put_vol
101:          bsr     _put_pan
102:          bra     main_lp
103:
104: fin:
105:          bsr     v_int_rmv
106:          bsr     int_se_play_init
107:          moveq   #$10,d1
108:          IOCS    _CRTMOD
109:          clr.w   -(sp)
110:          move.w  #14,-(sp)                       * ファンクションキー表示
111:          DOS     _CONCTRL
112:          move.w  #31,-(sp)
113:          clr.w   -(sp)
114:          move.w  #15,-(sp)                       * スクロール範囲設定
115:          DOS     _CONCTRL
116:          move.w  #17,-(sp)                       * カーソル表示
117:          DOS     _CONCTRL
118:          move.w  #$0000,-(sp)                    * キーバッファ初期化
119:          DOS     _KFLUSH
120:          lea     14(sp),sp
121: fin02:
122:          move.l  to_user_sp(a6),a1
123:          IOCS    _B_SUPER                        * ユーザーモード移行
124:          move.l  stack_buf(pc),sp
125:          DOS     _EXIT
126: *////////////////////////////////////////////////////////////
127: v_int_set:
128:          moveq   #1,d1
129:          lea     v_int(pc),a1
130:          IOCS    _VDISPST
131:          tst.l   d0
132:          bne     v_int_set_err
133:          rts
134:
135: v_int_rmv:
136:          move.w  SR,-(sp)
137:          ori.w   #$0700,SR
138:          suba.l  a1,a1
139:          IOCS    _VDISPST
140:          move.w  (sp)+,SR
141:          rts
142:
143: v_int:
144:          bclr.b  #5,$e88013                      *TIMER_A_INT_OFF
145:          andi.w  #$f8ff,SR
146:          movem.l d0-d7/a0-a6,-(sp)
147:          lea     work_top(pc),a6                 * a6:
148:          btst.b  #3,$e88015
149:          beq     @f
150:          bsr     int_se_play
151: @@:
152:          addq.w  #1,v_cnt(a6)
153:          bsr     _key_job
154:          movem.l (sp)+,d0-d7/a0-a6
155:          bset.b  #5,$e88013                      *TIMER_A_INT_ON
156:          rte
157:
158: v_int_set_err:
159:          pea     @f(pc)
160:          DOS     _PRINT
161:          bra     fin02
162: @@:
163:          dc.b    'V-DISP割り込みが使用中です',13,10,0
164:          .even
165: *////////////////////////////////////////////////////////////
166: *        OPMビジー検査
167: opm_busy_chk:
168: @@:      tst.b   $e90003
169:          bmi     @b
170:          rts
171: *////////////////////////////////////////////////////////////
172: zmsc_inst_chk:
173:          move.l  $8c.w,a0
174:          subq.w  #8,a0
175:          cmpi.l  #'ZmuS',(a0)+
176:          bne     zmsc_inst_err
177:          cmpi.w  #'iC',(a0)+
178:          bne     zmsc_inst_err
179:          move.w  (a0)+,d0
180:          move.w  d0,d1                           * d1:Ver_No.
181:          andi.w  #$ff0f,d0
182:          cmpi.w  #ZM_VER,d0
183:          bcs     zmsc_ver_err
184:          rts
185:
186: zmsc_inst_err:
187:          pea     ZMSC_INST_ERRMES01(pc)
188:          DOS     _PRINT
189:          bra     fin02
190:
191: zmsc_ver_err:
192:          pea     ZMSC_INST_ERRMES02(pc)
193:          DOS     _PRINT
194:          bra     fin02
195:
196: ZMSC_INST_ERRMES02:
197:          dc.b    'バージョン'
198:          ZM_VERs
199:          dc.b    '以降の'
200: ZMSC_INST_ERRMES01:
201:          dc.b    'Z-MUSICが常駐していません',13,10,0
202:          .even
203: *////////////////////////////////////////////////////////////
204: fm78_maskon:
205:          moveq.l #1,d2
206:          Z_MUSIC $3c
207:          move.l  a0,a5                          * a5:seq_wk_tbl base addr.
208:          Z_MUSIC $3a                            * a0:play_trk_tbl
209: @@:      move.b  (a0)+,d0
210:          bmi     fm78_maskon_fin
211:          lsl.w   #8,d0
212:          lea     $00(a5,d0.w),a4                * a4:
213:          cmpi.b  #$06,p_ch(a4)                  * FM7ch?
214:          beq     maskon
215:          cmpi.b  #$07,p_ch(a4)                  * FM8ch?
216:          beq     maskon
217:          bra     @b
218: maskon:
219:          move.b  #$00,p_se_mode(a4)
220:          bra     @b
221:
222: fm78_maskon_fin:
223:          rts
224: *////////////////////////////////////////////////////////////
225: *        初期化
226: int_se_play_init:
227:          st.b    zmse_seq_wk1+p_not_empty(a6)
228:          st.b    zmse_seq_wk2+p_not_empty(a6)
229:          move.b  #6,zmse_seq_wk1+p_ch(a6)
230:          move.b  #7,zmse_seq_wk2+p_ch(a6)
231:          OPMSET  #8,#6                          * キーオフ
232:          OPMSET  #8,#7                          * キーオフ
233:          rts
234:
235: *        コール
236: int_se_play:
237:          btst.b  #3,$e88015
238:          bne     @f
239:          rts
240: @@:
241:          lea     zmse_seq_wk1(a6),a5
242:          tst.b   p_not_empty(a5)
243:          bne     @f
244:          Z_MUSIC $58
245: @@:
246:          lea     zmse_seq_wk2(a6),a5
247:          tst.b   p_not_empty(a5)
248:          bne     @f
249:          Z_MUSIC $58
250: @@:
251:          OPMSET  #$14,#%0011_1010
252:          rts
253: *////////////////////////////////////////////////////////////
254: *                [in]    d0.b    効果音番号
255: *                        d1.b    パンポット         = $b3   P3(C)
256: *                                                   = $b1   P1(L)
257: *                                                   = $b2   P2(R)
258: *                        d2.b    ボリューム
259: int_se_play_R:
260:          movem.l d0-d7/a1-a3,-(sp)
261:          lea     DTBL_SEDATA(pc),a1
262:          andi.w  #$00ff,d0
263:          add.w   d0,d0
264:          adda.w  $00(a1,d0.w),a1                * a1:SE_DATA
265: *
266:          OPMSET  #8,#6                          * キーオフ
267:          OPMSET  #8,#7                          * キーオフ
268:          cmpi.w  #$0001,(a1)+
269:          beq     @f
270:          move.l  (a1)+,d0                       * d0:
271:          lea     (a1,d0.l),a2                   * a2:
272:          move.b  d1,4(a2)                   * MMLは'@_ v_ p_'で始まること
273:          move.b  d2,3(a2)
274:          lea     zmse_seq_wk1(a6),a3            * a3:
275:          clr.b   p_not_empty(a3)                * 演奏状態に
276:          move.l  a2,p_data_pointer(a3)          * set address
277:          bsr     zmse_seq_wk_init
278:          addq.w  #2,a1
279:          bra     @@f
280: @@:
281:          st.b    zmse_seq_wk1+p_not_empty(a6)
282:                                                 * 死亡状態に
283: @@:
284: *
285:          move.l  (a1)+,d0                       * d0:
286:          lea     (a1,d0.l),a2                   * a2:
287:          move.b  d1,4(a2)                   * MMLは'@_ v_ p_'で始まること
288:          move.b  d2,3(a2)
289:          lea     zmse_seq_wk2(a6),a3            * a3:
290:          clr.b   p_not_empty(a3)                * 演奏状態に
291:          move.l  a2,p_data_pointer(a3)          * set address
292:          bsr     zmse_seq_wk_init
293:          movem.l (sp)+,d0-d7/a1-a3
294:          rts
295: *-------------------------------------------------------------
296: *        zmse_seq_wkの初期化
```

▶大学に通うようになって，月2回のペースで秋葉原に行っている。アキバマスターになる日も近い。
鳩山　智之(18)神奈川県

```
297: *                  [in]    a3      zmse_seq_wk
298: zmse_seq_wk_init:
299:            move.l  #$0001_0001,(a3)
300:            moveq   #0,d0
301:            move.b  d0,p_fo_mode(a3)
302:            move.b  d0,p_non_off(a3)
303:            move.l  d0,p_rpt_cnt(a3)
304:            move.l  d0,p_rpt_cnt+4(a3)
305:            move.l  d0,p_bend_sw(a3)
306:            move.l  d0,p_port_flg(a3)
307:            move.b  d0,p_rpt_last?(a3)
308:            move.w  d0,p_pb_add(a3)
309:            move.l  d0,p_arcc_flg(a3)
310:            move.l  d0,p_pmod_flg(a3)
311:            move.w  d0,p_seq_flag(a3)
312:            move.l  d0,p_tie_pmod(a3)
313:            moveq   #-1,d0
314:            move.w  d0,p_pmod_mode(a3)
315:            move.b  d0,p_marker(a3)
316:            move.w  d0,p_note(a3)
317:            move.b  d0,p_se_mode(a3)
318:            rts
319: *///////////////////////////////////////////////////////////////
320: _key_job:
321:            IOCS    _B_KEYSNS
322:            tst.l   d0
323:            bne     @f
324:            rts
325: @@:
326:            IOCS    _B_KEYINP
327:            andi.w  #$ff00,d0
328:            cmpi.w  #$4600,d0                       * [+]
329:            beq     _kj_SENO_P
330:            cmpi.w  #$4200,d0                       * [-]
331:            beq     _kj_SENO_M
332:            cmpi.w  #$4400,d0                       * [8]
333:            beq     _kj_VOL_P
334:            cmpi.w  #$4C00,d0                       * [2]
335:            beq     _kj_VOL_M
336:            cmpi.w  #$4700,d0                       * [4]
337:            beq     _kj_PAN_L
338:            cmpi.w  #$4800,d0                       * [5]
339:            beq     _kj_PAN_C
340:            cmpi.w  #$4900,d0                       * [6]
341:            beq     _kj_PAN_R
342:            cmpi.w  #$4E00,d0                       * [ENTER]
343:            beq     _kj_SEREQ
344:            rts
345:
346: _kj_SENO_P:
347:            cmpi.w  #HM_SE-1,se_no(a6)
348:            beq     @f
349:            addq.w  #1,se_no(a6)
350: @@:
351:            rts
352:
353: _kj_SENO_M:
354:            tst.w   se_no(a6)
355:            beq     @f
356:            subq.w  #1,se_no(a6)
357: @@:
358:            rts
359:
360: _kj_VOL_P:
361:            tst.w   se_vol(a6)
362:            beq     @f
363:            subq.w  #1,se_vol(a6)
364: @@:
365:            rts
366:
367: _kj_VOL_M:
368:            cmpi.w  #127-85,se_vol(a6)
369:            beq     @f
370:            addq.w  #1,se_vol(a6)
371: @@:
372:            rts
373:
374: _kj_PAN_L:
375:            move.w  #$b1,se_pan(a6)
376:            rts
377:
378: _kj_PAN_C:
379:            move.w  #$b3,se_pan(a6)
380:            rts
381:
382: _kj_PAN_R:
383:            move.w  #$b2,se_pan(a6)
384:            rts
385:
386: _kj_SEREQ:
387:            move.w  se_no(a6),d0
388:            move.w  se_pan(a6),d1
389:            move.w  se_vol(a6),d2
390:            bsr     int_se_play_R
391:            rts
392:
393: _put_seno:
394:            B_LOCA  9,0
395:            move.w  se_no(a6),d1
396:            bra     hex_put_1b
397:
398: _put_vol:
399:            B_LOCA  9,1
400:            move.w  se_vol(a6),d1
401:            bra     hex_put_1b
402:
403: _put_pan:
404:            move.w  se_pan(a6),d0
405:            subi.w  #$b1,d0
406:            add.w   d0,d0
407:            move.w  d0,d1
408:            add.w   d0,d0
409:            add.w   d1,d0
410:            DISPLAY @f(pc,d0.w),3,8,2,6
411:            rts
412: @@:
```

```
413:            dc.b    'Left  '
414:            dc.b    'Right '
415:            dc.b    'Center'
416:            .even
417:
418: *          [in]    d1
419: hex_put_1b:
420:            move.l  d1,d2
421:            lsr.w   #4,d1
422:            andi.w  #$000f,d1
423:            B_PUTC  @f(pc,d1.w)
424:            move.l  d2,d1
425:            andi.w  #$000f,d1
426:            B_PUTC  @f(pc,d1.w)
427:            rts
428: @@:
429:            dc.b    '0123456789ABCDEF'
430: *///////////////////////////////////////////////////////////////
431:                    .data
432:                    .even
433: mes_SNO:
434:            dc.b    'SE_No.: $00'
435: mes_VOL:
436:            dc.b    'Volume: $00'
437: mes_PAN:
438:            dc.b    'Panpot: Center'
439:            .even
440: DTBL_SEDATA:
441: @@:        dc.w    SE_0-@b
442:            dc.w    SE_1-@b
443:            dc.w    SE_2-@b
444:            dc.w    SE_3-@b
445:            dc.w    SE_4-@b
446:            dc.w    SE_5-@b
447: *
448: SE_0:
449:            dc.b    $00,$01                         * 演奏総トラック数
450:            dc.b    $00,$00,$00,$02                 * 演奏データまでのｵﾌｾｯﾄ
451:            dc.b    $00,$07                         * 演奏絶対チャンネル
452:                                                    * 以下演奏データ
453:            dc.b    $A0,$C8                         * @200
454:            dc.b    $B6,$00                         * v127
455:            dc.b    $B3                             * p3
456:            dc.b    $D1,$00,$00,$00,$00             * @k0
457:            dc.b    $BB,$00                         * @m
458:            dc.b    $82                             * @o
459:            dc.b    $C1,$CF,$02                     * |:2
460:            dc.b    $29,$01,$FF                     * o2f*1
461:            dc.b    $28,$01,$FF                     * o2e*1
462:            dc.b    $26,$01,$FF                     * o2d*1
463:            dc.b    $24,$01,$FF                     * o2c*1
464:            dc.b    $23,$01,$FF                     * o1b*1
465:            dc.b    $21,$01,$FF                     * o1a*1
466:            dc.b    $1F,$01,$FF                     * o1g*1
467:            dc.b    $1D,$01,$FF                     * o1f*1
468:            dc.b    $AB,$0A                         * _10
469:            dc.b    $C2,$00,$1F                     * :|
470:            dc.b    $FF                             * End of Data
471: SE_1:
472:            dc.b    $00,$01,$00,$00,$00,$02,$00,$07
473:            dc.b    $A0,$C8,$B6,$00,$B3,$D1,$00,$00
474:            dc.b    $00,$00,$BB,$00,$82,$AB,$08,$C1
475:            dc.b    $CF,$0C,$3C,$01,$FF,$35,$01,$FF
476:            dc.b    $C2,$00,$0B,$FF
477: SE_2:
478:            dc.b    $00,$01,$00,$00,$00,$02,$00,$07
479:            dc.b    $A0,$C8,$B6,$02,$B3,$D1,$00,$00
480:            dc.b    $00,$00,$BB,$00,$82,$5D,$12,$12
481:            dc.b    $FF,$00
482: SE_3:
483:            dc.b    $00,$01,$00,$00,$00,$02,$00,$07
484:            dc.b    $A0,$C8,$B6,$02,$B3,$D1,$00,$00
485:            dc.b    $00,$00,$BB,$00,$82,$C1,$CF,$05
486:            dc.b    $4F,$01,$FF,$37,$01,$FF,$4F,$01
487:            dc.b    $FF,$43,$01,$FF,$AB,$05,$C2,$00
488:            dc.b    $13,$FF
489: SE_4:
490:            dc.b    $00,$01,$00,$00,$00,$02,$00,$07
491:            dc.b    $A0,$C8,$B6,$02,$B3,$D1,$00,$00
492:            dc.b    $00,$00,$BB,$00,$82,$54,$01,$FF
493:            dc.b    $57,$01,$FF,$54,$01,$FF,$59,$01
494:            dc.b    $FF,$54,$01,$FF,$5C,$01,$FF,$FF
495: SE_5:
496:            dc.b    $00,$02,$00,$00,$00,$08,$00,$06
497:            dc.b    $00,$00,$00,$22,$00,$07,$A0,$C8
498:            dc.b    $B6,$00,$B3,$D1,$00,$00,$00,$00
499:            dc.b    $BB,$00,$82,$C1,$CF,$02,$E0,$18
500:            dc.b    $00,$18,$00,$18,$00,$00,$00,$40
501:            dc.b    $00,$01,$C2,$00,$11,$FF,$A0,$C8
502:            dc.b    $B6,$00,$B3,$D1,$00,$00,$00,$00
503:            dc.b    $BB,$00,$82,$C1,$CF,$02,$E0,$24
504:            dc.b    $00,$18,$00,$18,$00,$00,$00,$40
505:            dc.b    $00,$01,$C2,$00,$11,$FF
506: *///////////////////////////////////////////////////////////////
507:                    .offset 0
508: to_user_sp:        ds.l    1
509: v_cnt:             ds.w    1
510: se_no:             ds.w    1
511: se_vol:            ds.w    1
512: se_pan:            ds.w    1
513: zmse_seq_wk1:      ds.b    256
514: zmse_seq_wk2:      ds.b    256
515: work_size:
516:                    .bss
517:                    .even
518: work_top:          ds.b    work_size
519: *///////////////////////////////////////////////////////////////
520:                    .stack
521:                    .even
522:                    ds.l    1024
523: stack_buf:
524:                    ds.l    1
525: *///////////////////////////////////////////////////////////////
526: end_of_prog:
527:                    .end    start
528:
```

残響処理の高速化

FFTを使った畳み込み演算

Nobata Hideaki 野畠 英明

残響処理に威力を発揮するZPLKの畳み込み演算
ただ，演算に膨大な時間がかかるのが困りものだ
高速フーリエ変換アルゴリズムを利用して一気に高速化してみよう

Z-MUSICに付属のツールZVT（またはZPLK）には「畳み込み」という機能があります。ところで，「畳み込み」ってなにか知ってますか？　使ってみたことはありますか？　たぶん以前の特集で西川氏が解説されてましたから，知っている人は多いと思いますが，実際に使ってみた人は少ないんじゃないでしょうか。その理由としては，使いこなすことの難しさもさることながら，非常に処理時間がかかるということにもあると思います。

では，なんとか処理を速くすることはできないのでしょうか。そういえば，上記の西川氏の記事で高速フーリエ変換（FFT）を使う方法が簡単に紹介されてました。でも，実際にそのアルゴリズムでやってくれるプログラムは提示されてません。というわけで，FFTを用いた畳み込みについてのもう少し突っ込んだ解説と簡単なサンプルを紹介します。

畳み込み

一応，畳み込みについて軽くおさらいしておきましょう。

入力信号をx(t)，あるシステムのインパルス応答をh(t)，出力信号をy(t)とすると，

$$y(t)=x(t)h(t)$$
$$=\int_{-\infty}^{\infty}x(t)h(t-\tau)d\tau$$

という関係が成り立ちます。ちょっとわかりにくいかもしれませんね。

具体的な例でいうと，x(t)がエフェクトをかける前の音で，h(t)がホールの残響特性を表すインパルスデータ，y(t)がホールの残響のエフェクトがかかった音といったふうになります（別にx(t)とh(t)とを逆にしても同じことなのですが，便宜上そういうことにしておきます）。

さて，実際に扱うのはPCMデータですから，離散的で有限な信号です。というわけで，先の式を長さNの離散信号として書き直すと，

$$y(n)=\sum_{k=0}^{N-1}x(k)h(n-k)$$

となります。この式を真面目に計算してるのがZVTです。

ここで，この式をフーリエ変換（時間の関数を周波数の関数に変換）します。すると，面倒くさい計算が単なる積になってしまうのです。

$$Y(k)=X(k)H(k)$$

どうです。圧倒的に演算量が削減されることがわかりますね。あとは，逆フーリエ変換で元に戻せばOKです。時間領域と周波数領域での畳み込みの違いを図1に示しておきます。

ここで，注意しておきたいのは，周波数の関数になったときには長さNで循環しているということです。つまり，

$$Y(k)=Y(k+Ni)$$

i:任意の整数

なんですね。これを考えてないととんでもないことになる……かも。

さあ，これでめでたしめでたし……というわけにはいきません。というのもフーリエ変換自体が畳み込みのような計算なのです。N点離散フーリエ変換（DFT）の式は，

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{-j(2\pi/N)kn}$$

$$(j=\sqrt{-1})$$ *1

で表されますが，畳み込みとよく似ていることがわかるでしょう。

*1）あれ，虚数単位ってｉじゃないのって人もいるかもしれませんね。確かに本当はｉなんでしょうけど，電気屋さんはｊと書きます。というのも，ｉはよく電流を表す記号として使うからだったりします。

図1　畳み込みの概念図

高速フーリエ変換（FFT）

そこで出てくるのがFFTです。ここでFFTというのはもっともポピュラーと思われるクーリー・チューキーのアルゴリズムを用いるものです（バリエーションはいろいろあるようです）。

DFTの式の，

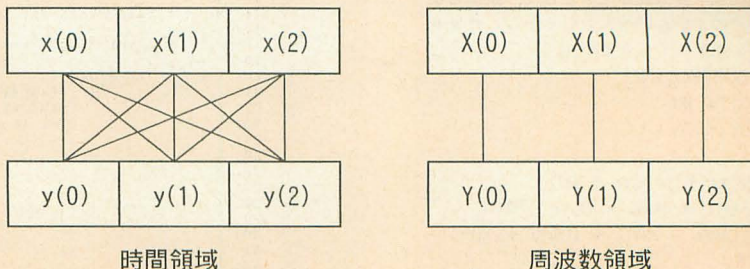
$$e^{-j(2\pi/N)kn}$$

の部分を書き下すと，

$$\cos(2\pi/N)kn-j\sin(2\pi/N)kn$$

となります。たとえば，4点とか8点とかそういう小さい長さのDFTを実際にどういう係数になるか代入してみてください。なんだか周期的になってませんか。

FFTはその周期性を利用して，問題をより小さな問題へと分割する，分割統治法という手法を用いて演算量を削減します。まず，もっとも簡単な2点DFTを考えます。2点DFTは次のようになります。

$$X(0)=x(0)w^0+x(1)w^0$$
$$X(1)=x(0)w^0+x(1)w^1$$
$$w=e^{-j(2\pi/2)}$$

w^0は1，w^1は−1ですから，さらに次のようになります。

$$X(0)=x(0)+x(1)$$
$$X(1)=x(0)-x(1)$$

それでは4点DFTを考えてみましょう。4点DFTは次のようになります。

$$X(0)=x(0)w^0+x(1)w^0+x(2)w^0+x(3)w^0$$
$$X(1)=x(0)w^0+x(1)w^1+x(2)w^2+x(3)w^3$$
$$X(2)=x(0)w^0+x(1)w^2+x(2)w^4+x(3)w^6$$
$$X(3)=x(0)w^0+x(1)w^3+x(2)w^6+x(3)w^9$$
$$w=e^{-j(2\pi/4)}$$

これも2点DFTのときと同様に簡単化してみます。すると，

$$X(0)=x(0)+x(1)+x(2)+x(3)$$
$$X(1)=x(0)-jx(1)-x(2)+jx(3)$$
$$X(2)=x(0)-x(1)+x(2)-x(3)$$
$$X(3)=x(0)+jx(1)-x(2)-jx(3)$$

のようになります。ここで，上式を奇数番目と偶数番目に分けて並び替えてやります。

$$X(0)=x(0)+x(2)+x(1)+x(3)$$
$$X(2)=x(0)+x(2)-x(1)-x(3)$$
$$X(1)=x(0)-x(2)-jx(1)+jx(3)$$
$$X(3)=x(0)-x(2)+jx(1)-jx(3)$$

なんか和と差で分けると，まとめることができそうですね。えいやっとまとめると，

$$X(0)=g(0)+g(1)$$
$$X(2)=g(0)-g(1)$$
$$X(1)=h(0)-jh(1)$$
$$X(3)=h(0)+jh(1)$$

$$g(0)=x(0)+x(2)$$
$$g(1)=x(1)+x(3)$$
$$h(0)=x(0)-x(2)$$
$$h(1)=x(1)-x(3)$$

となってかなり簡単になりました。

これをよく見るとなんだか2点DFTによく似ていませんか？　X(1)とX(3)はjがかかってますが，これは奇数番目と偶数番目との位相のずれの分です（この場合はひとつ分ずれてるわけです）。これまでの4点DFTの分割の様子を図にしたものが図2です。

点数が多い場合でも同様にして順次2分割していけばいいわけですが，この例のように順次2分割していく場合は$N=2^c$である必要があります。

このように2分割していくものを基数2のFFTといいます。これは2点DFTが基本になっていて，2点DFTを繰り返し演算することになります。これ以外にも基数4のものや基数8のものがよく使われます。

基数を大きくしてなにがよいかというと，分割の回数が減るわけです。たとえば，基数4のFFTは基数2のFFTに比べて分割が半分で済むことになり，位相回転の分を考えあわせても3/4の乗算回数で済むことになります。ただし，基数を大きくすると，FFTの点数の自由度が少なくなりますから，基数の大きいものでできるだけ演算し，半端なところは基数の小さいもので演算するというのがベストでしょう。

より詳しい説明は専門書を参照してください。

実数データに対する工夫

PCMデータはいうまでもなく実数データで，虚数なんて関係ありません。なのに，フーリエ変換は複素数を前提としています。これでは無駄があるような気がしませんか？　実際，実数（虚数部が0と考える）をフーリエ変換すると，その出力は実数部が偶関数（$f(x)=f(-x)$），虚数部が奇関数（$f(x)=-f(-x)$）となります（エルミート対称）。この対称性をなんとか生かすことができれば高速化が図れそうです。

ところで，一般の関数を偶関数と奇関数に分離してみましょう。

$$f(x)=f(x)/2+f(x)/2$$
$$=(f(x)+f(-x))/2+(f(x)-f(-x))/2$$

ここで，$(f(x)+f(-x))/2$の部分はxを−xと置き換えても変わらないから偶関数，$(f(x)-f(-x))/2$の部分はxを−xと置き換えると符号が反転するので奇関数ですね。

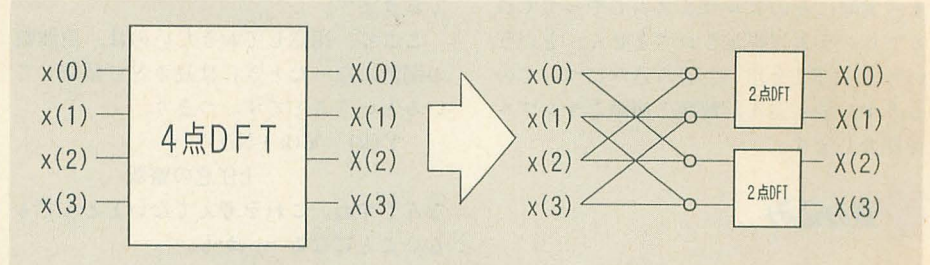
これを利用して，2つの実数データを同時にFFTで処理することができます。ひとつ目のデータをf(n)，2つ目のデータをg(n)とし，次のような関数を考えます。

$$x(n)=f(n)+jg(n)$$

フーリエ変換して，

$$X(k)=F(k)+jG(k)$$

となります。F(k)の実数部をFr(k)，虚数部をFi(k)とし，またG(k)の実数部をGr(k)，虚数部をGi(k)とすれば，

図2　DFTの分割

リスト1　FFT.C

```
 1: /*
 2:                     ＦＦＴ関係
 3:
 4:                     $Id: fft.c,v 1.1.1.1 1995/01/26 13:55:52 nova Exp $
 5: */
 6:
 7: #include            <stdio.h>
 8: #include            <stdlib.h>
 9: #include            <math.h>
10: #include            <sys/types.h>
11: #include            "convo.h"
12:
13: /*
14:           ＦＦＴ（基数２）を行う（ inv ≠ 0 なら逆ＦＦＴになる）
15: */
16: void fft2(COMPLEX *xr, COMPLEX *xi, uint n, uint nu, int inv)
17: {
18:           uint l, n2, nul, k;
19:
20:           if (inv) {
21:                     /* 逆転する */
22:                     uint n2 = n / 2;
23:                     for (k = 1; k < n2; k++) {
24:                               COMPLEX tmp;
25:                               tmp = xr[k];
26:                               xr[k] = xr[n-k];
27:                               xr[n-k] = tmp;
28:                               tmp = xi[k];
29:                               xi[k] = xi[n-k];
30:                               xi[n-k] = tmp;
31:                     }
32:           }
33:
```

複素数の積

複素数の和や差は問題ないとして，積はちょっとわかりにくいという人もいるかもしれませんので少し補足しておきます。

2つの複素数XとYを，

$$X = Xr + jXi$$
$$Y = Yr + jYi$$

とすると，

$$X \times Y=(Xr + jXi)(Yr + jYi)$$
$$=XrYr + jXrYi + jXiYr - XiYi$$
$$=(XrYr - XiYi) + j(XrYi + XiYr)$$

となります。

$$X(k)=Fr(k)+jFi(k)+j(Gr(k)+jGi(k))$$
$$=Fr(k)-Gi(k)+j(Fi(k)+Gr(k))$$
と変形できます。

したがって，X(k)の実数部をXr(k)，虚数部をXi(k)とすれば，

$$Xr(k)=Fr(k)-Gi(k)$$
$$Xi(k)=Fi(k)+Gr(k)$$

となり，先の一般の関数を偶関数と奇関数に分ける方法により2つの実数データのフーリエ変換の結果を得ることができるわけです。

たとえばFr(k)は，Xr(k)の偶関数成分ですから，

$$Fr(k)=(Xr(k)+Xr(-k))/2$$

です。ここで，Xr(−k)というのはXr(N−k)のことです（先に述べたように循環しているから）。同様に，

$$Fi(k)=(Xi(k)-Xi(-k))/2$$
$$Gr(k)=(Xi(k)+Xi(-k))/2$$
$$Gi(k)=(Xr(-k)-Xr(k))/2$$

となります。

このようにしてフーリエ変換を約2倍の効率で行うことができるわけです。

同様の手法とFFTの分割の発想で，ひとつの実数データでも約2倍の効率で行うこともできます（計算はやや複雑になる）。

逆フーリエ変換

フーリエ変換をしたあとは元に戻すために逆フーリエ変換をする必要がありますが，これはフーリエ変換を流用することができます。

逆DFTの式は，

$$x(n)=1/N\sum_{k=0}^{N-1}X(k)e^{j(2\pi/N)kn}$$

ですが，この両辺の複素共役をとります。すると，

$$x^*(n)=(1/N)\sum_{k=0}^{N-1}X(k)e^{-j(2\pi/N)kn}$$

となりますが，これは$X^*(k)$にDFTをかけて1/Nしたものになっています。

というわけで，複素共役をとってDFTをかけ，1/Nして，再び複素共役をとれば逆DFTになります。ただし，今回はPCMデータを扱っているわけですから最後の複素共役をとる処理は省くことができます。

サンプルプログラム

さて，数式の説明だけじゃさっぱりわからないという人が多いでしょうから，サンプルを見てみましょう。

図3にサンプルプログラムの流れを示し

```
34:         /* ＦＦＴメイン */
35:         n2 = n / 2;
36:         nu1 = nu - 1;
37:
38:         for (l = 1; l <= nu; l++) {
39:                 uint k = 0;
40:                 do {
41:                         uint i;
42:                         for (i = 1; i <= n2; i++) {
43:                                 uint kn2;
44:                                 uint arg;
45:                                 COMPLEX c, s;
46:                                 COMPLEX tr, ti;
47:                                 arg = intBitR(k >> nu1, nu);
48:                                 c = cs[arg];
49:                                 s = sn[arg];
50:                                 kn2 = k + n2;
51:                                 tr = xr[kn2] * c + xi[kn2] * s;
52:                                 ti = xi[kn2] * c - xr[kn2] * s;
53:                                 xr[kn2] = xr[k] - tr;
54:                                 xi[kn2] = xi[k] - ti;
55:                                 xr[k] += tr;
56:                                 xi[k] += ti;
57:                                 k++;
58:                         }
59:                         k += n2;
60:                 } while (k < n);
61:                 nu1--;
62:                 n2 /= 2;
63:         }
64:
65:         /* 並べ直す */
66:         for (k = 0; k < n; k++) {
67:                 uint i;
68:                 COMPLEX tr, ti;
69:                 i = intBitR(k, nu);
70:                 if (i <= k) continue;
71:                 tr = xr[k];
72:                 ti = xi[k];
73:                 xr[k] = xr[i];
74:                 xi[k] = xi[i];
75:                 xr[i] = tr;
76:                 xi[i] = ti;
77:         }
78:
79:         if (inv) {
80:                 /* 要素数で割る */
81:                 COMPLEX nd = (COMPLEX) n;
82:                 for (k = 0; k < n; k++) {
83:                         *xr++ /= nd;
84:                 }
85:         }
86: }
```

リスト2

```
 1: /*
 2:                         ＦＦＴを使用した畳み込み積分を行う
 3:
 4:                         convo 簡易バージョン
 5:
 6:                         $Id: main.c,v 1.1.1.1 1995/01/26 13:55:50 nova Exp $
 7: */
 8:
 9: #include <stdio.h>
10: #include <stdlib.h>
11: #include <unistd.h>
12: #include <math.h>
13: #include <string.h>
14: #include "convo.h"
15:
16: #define TITLE "Convolution calculator (簡易版) (c)1995 NOVA¥n"
17:
18: short   *pcm1, *pcm2, *pcm3;
19: COMPLEX *xr, *xi;
20: float   *sn, *cs;
21:
22: void usage()
23: {
24:         fprintf( stderr,
25:                 "Usage: convo input-pcm1 input-pcm2 output-pcm¥n");
26: }
27:
28: void error()
29: {
30:         perror("convo");
31:         exit(1);
32: }
33:
34: void printMsg(char *string)
35: {
36:         fprintf(stderr, string);
37: }
38:
39: void *getMem(size_t n, size_t s)
40: {
41:         void *tmp;
42:
43:         if ((tmp = malloc(n * s)) == NULL) error();
44:         return tmp;
45: }
46:
47: void *getMemC(size_t n, size_t s)
48: {
49:         void *tmp;
50:
51:         if ((tmp = calloc(n, s)) == NULL) error();
52:         return tmp;
53: }
54:
```

ます。基本的にこれまでの説明通りのオーソドックスな作りになっています。が，これまでの説明にないちょっとした工夫もしてあります。

まず，三角関数は基本的に重い関数なのでテーブル参照にしてあります。また，sinとcosは$\pi/2$だけ位相がずれている関数で形は同じですから，テーブルをある程度共有させてメモリ消費を多少抑えてあります。あと，逆FFTですが，説明した方法と違ってデータを逆転（X(1)がX(N−1)に，X(N−1)がX(1)に入れ替わる）しています。たぶん，このほうが速いでしょう。

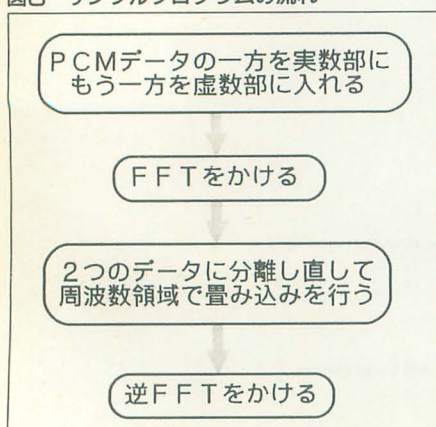
それから，FFTの関数ですが，中間結果を上書きすることで余分なバッファを不要にしています（この辺のことは説明しませんでしたが，FFTでは基本的な手法だったりします)。この処理の関係でビット反転する関数が必要で，それはCで書くのは面倒なのでアセンブラで書いてあります。工夫はそんなところですが，手抜きもあります。それは，入力/出力とも16ビットリニアPCMデータとなっている点です。AD PCMでは処理できないのでこういう仕様になってますが，いちいち前もって変換するのが面倒だという人は自動で変換するように拡張するといいでしょう。ちなみに16ビットリニアPCMデータとAD PCMデータの変換はZ-MUSICのライブラリに関数がありますので，それを利用すると便利かと思います。

それでは，実際に使ってみましょう。このプログラムは，

gcc v1.28
has v3.09
hlk v3.01
libc-1.1.32A
GNU make v3.62 X6_12

などを用いてコンパイルできることを確認してあります。各ツールの細かいバージョ

図3　サンプルプログラムの流れ

PCMデータの一方を実数部に
もう一方を虚数部に入れる
↓
FFTをかける
↓
2つのデータに分離し直して
周波数領域で畳み込みを行う
↓
逆FFTをかける

```
 55: size_t readFile(char *filename, void **ptr)
 56: {
 57:         FILE *inp;
 58:         size_t size;
 59:
 60:         if ((inp = fopen(filename, "rb")) == NULL) error();
 61:         size = filelength(fileno(inp));
 62:         if ((*ptr = (void *) malloc(size)) == NULL) error();
 63:         fread(*ptr, 1, size, inp);
 64:         fclose(inp);
 65:
 66:         return size;
 67: }
 68:
 69: /*
 70:         ＳＩＮ・ＣＯＳテーブル作成
 71: */
 72: void makeSinCosTable(size_t n)
 73: {
 74:         size_t i, n4;
 75:         float *ptr, *ptr2;
 76:         double pi2 = M_PI * 2 / n;
 77:
 78:         n4 = n / 4;
 79:         ptr = sn;
 80:         for (i = 0; i <= n4; i++) {
 81:                 *ptr++ = sin(pi2 * i);
 82:         }
 83:         ptr2 = --ptr;
 84:         for (i = 0; i < n4; i++) {
 85:                 *ptr2++ = *ptr--;
 86:         }
 87:         n4 *= 2;
 88:         for (i = 0; i < n4; i++) {
 89:                 *ptr2++ = -(*ptr++);
 90:         }
 91:         n4 /= 2;
 92:         ptr = sn;
 93:         ptr2 = sn + n;
 94:         for (i = 0; i < n4; i++) {
 95:                 *ptr2++ = *ptr++;
 96:         }
 97: }
 98:
 99: /*
100:         ＰＣＭデータを複素数に変換して格納する
101: */
102: void PcmToComplex(COMPLEX *xr, short *pcm, size_t pcm_size)
103: {
104:         size_t  i;
105:
106:         for (i = 0; i < pcm_size; i++)
107:                 *xr++ = *pcm++;
108: }
109:
110: /*
111:         複素数をＰＣＭデータに変換して格納する
112: */
113: void ComplexToPcm(short *pcm, COMPLEX *xr, size_t pcm_size)
114: {
115:         size_t i;
116:         COMPLEX max;
117:
118:         max = 0x10000;
119:         for (i = 0; i < pcm_size; i++) {
120:                 *pcm++ = (*xr++) / max;
121:         }
122: }
123:
124: /*
125:         分解及び畳み込み処理
126: */
127: void sepaConvo(unsigned int n)
128: {
129:         unsigned int i, n2;
130:         n2 = n / 2;                                     /* エルミート対称となるから半分
てよい */
131:         xr[n] = xr[0];
132:         xi[n] = xi[0];
133:         for (i = 0; i <= n2; i++) {
134:                 COMPLEX re1, im1, re2, im2;
135:                 /* ２つの関数に分解 */
136:                 re1 = (xr[i] + xr[n-i]) / 2;
137:                 im1 = (xi[i] - xi[n-i]) / 2;
138:                 re2 = (xi[i] + xi[n-i]) / 2;
139:                 im2 = (xr[n-i] - xr[i]) / 2;
140:                 /* 複素数の乗算 */
141:                 xr[n-i] = xr[i] = re1 * re2 - im1 * im2;
142:                 xi[i] = re1 * im2 + im1 * re2;
143:                 xi[n-i] = -xi[i];
144:         }
145: }
146:
147: /*
148:         メインルーチン
149: */
150: int main(int argc, char *argv[])
151: {
152:         size_t pcm1_size, pcm2_size, pcm_size;
153:         unsigned int n, nu;
154:         FILE *out;
155:
156:         fprintf(stderr, TITLE);
157:
158:         if (argc != 4) {
159:                 usage();
160:                 printMsg("引数の数が違います¥n");
161:                 exit(1);
162:         }
163:
164:         pcm1_size = readFile(argv[1], (void **) &pcm1);
165:         pcm1_size /= sizeof(short);
```

ンはあまり関係ないと思いますが，XCのライブラリだとそのままではコンパイルできないと思います。たぶん，多少の変更で動くとは思いますけど。

あと，X68030でFPUを登載してある機種ではMakefileのCFLAGSのコメントになっているほうを有効にするとFPUを直接ドライブするコードが出力できます。X68030以外でFPUボードなどを登載している場合は，-m68020を外して，さらにリンク時のオプションに-lsuperをつけ加えると直接ドライブするコードが出力できるはずですが試してません。

Z-MUSIC本についているPCMデータで試すことにします。なんでもいいんですが，RAP¥GET3.PCM と EFFECTS¥WATER.PCMを畳み込んでみます。

その2つのPCMデータがカレントディレクトリにあると仮定して，

A>zvt -c GET3.PCM GET3.P16

A>zvt -c WATER.PCM WATER.P16

A>convo GET3.P16 WATER.P16 TEST.P16

で畳み込んだPCMデータがTEST.P16に書き出されます。この16ビットリニアPCMデータはそのままでは再生できないので，ZVTなどで聞いてみてください。元になった2つのPCMが混じったような感じになっていると思います（音量が小さくなっちゃうので，2倍くらいにしたほうが聞きやすいかもしれません）。

最後に

かなり飛ばしてFFTによる畳み込みについて書いてみましたが，ある程度理解できたでしょうか。ところどころはしょってるところもありますから，わかりにくい点もあるかと思います。そういうところはサンプルソースや参考文献を参照したりしてみてください。あと，残念なのが2つのPCMが極端に長さが違う場合の対処法を示せなかったことです。本当はそのあたりも解説したかったのですが。

また，ここで示した方法のほかにも高速算法はいろいろなバリエーションや手法があります。たとえば，数論変換やFFTの代わりにWFTなどを用いるとか。興味があればいろいろ調べてみるのもいいでしょう。

参考文献

高速フーリエ変換，科学技術出版社

ディジタル信号処理，昭晃堂

岩波講座情報科学-18数値計算，岩波書店


```
166:
167:            pcm2_size = readFile(argv[2], (void **) &pcm2);
168:            pcm2_size /= sizeof(short);
169:
170:            pcm_size = pcm1_size + pcm2_size - 1;
171:            nu = (unsigned int) ceil(log((double) pcm_size) / M_LN2);
172:            n = 1 << nu;
173:
174:            xr = (COMPLEX *) getMemC(n + 1, sizeof(COMPLEX));
175:            xi = (COMPLEX *) getMemC(n + 1, sizeof(COMPLEX));
176:            sn = (float *) getMem(n + n / 4, sizeof(float));
177:            cs = sn + n / 4;
178:
179:            PcmToComplex(xr, pcm1, pcm1_size);
180:            PcmToComplex(xi, pcm2, pcm2_size);
181:            free(pcm1);
182:            free(pcm2);
183:
184:            printMsg("ＳＩＮ・ＣＯＳテーブル作成スタート¥n");
185:            makeSinCosTable(n);
186:
187:            printMsg("ＰＣＭ１とＰＣＭ２のＦＦＴスタート¥n");
188:            fft2(xr, xi, n, nu, 0);
189:
190:            printMsg("ＰＣＭ１とＰＣＭ２への分解及び");
191:            printMsg("畳み込み演算スタート¥n");
192:            sepaConvo(n);
193:
194:            printMsg("処理済みＰＣＭの逆ＦＦＴスタート¥n");
195:            fft2(xr, xi, n, nu, 1);
196:
197:            free(xi);
198:
199:            pcm3 = (short *) getMem(pcm_size, sizeof(short));
200:
201:            ComplexToPcm(pcm3, xr, pcm_size);
202:
203:            /* ファイルを書き出す */
204:            if ((out = fopen(argv[3], "wb")) == NULL) error();
205:            fwrite(pcm3, sizeof(short), pcm_size, out);
206:            fclose(out);
207:
208:            free( xr );
209:            free( pcm3 );
210:
211:            printMsg("終了しました¥n");
212:
213:            return 0;
214: }
```

リスト3 intBit.S

```
 1: *
 2: *          ビット反転（１ビット単位）
 3: *
 4: *          $Id: intBitR.s,v 1.1.1.1 1995/01/26 13:55:54 nova Exp $
 5: *
 6:            .globl  _intBitR
 7: _intBitR:
 8:            move.l  4(sp),d2
 9:            move.l  8(sp),d1
10:            subq.w  #1,d1
11:            moveq.l #0,d0
12: @@:
13:            lsr.l   #1,d2
14:            addx.l  d0,d0
15:            dbra    d1, @b
16:            rts
```

リスト4 CONVO.H

```
 1: /*
 2:                 convo ヘッダファイル
 3:
 4:                 $Id: convo.h,v 1.1.1.1 1995/01/26 13:55:48 nova Exp $
 5: */
 6:
 7: typedef float   COMPLEX;
 8:
 9: extern float *sn, *cs;
10:
11: extern int intBitR(unsigned int, unsigned int);
12: extern void fft2(COMPLEX *, COMPLEX *, unsigned int, unsigned int, int);
```

リスト5 Makefile

```
 1: #
 2: #          Convo: convolution program
 3: #
 4: #          $Id: Makefile,v 1.1.1.1 1995/01/26 13:55:46 nova Exp $
 5: #
 6: TARGET   = convo.x
 7: OBJS     = main.o fft.o intBitR.o
 8: CFLAGS   = -Wall -O -fall-bsr
 9: #CFLAGS = -Wall -m68020 -m68881 -O -fall-bsr
10:
11: .PHONY: clean
12:
13: $(TARGET): $(OBJS)
14:            $(CC) -o $@ $^
15:
16: clean:
17:            $(RM) $(TARGET) $(OBJS)
18:
19: # header dependency
20: main.c: convo.h
21: fft.c: convo.h
```

[特集] SoundEffects

PCM音を分解する

逆フーリエ変換による周波数解析

Taki Yasushi 瀧 康史

単純なPCMデータ加工だけでは音量しか変化しない
もっともっと多彩なフィルタ処理の前段階として
周波数帯ごとの処理をするための中間フォーマットを作成する

標本化

世の中にはいろんな意味で，波形として表すことができるものがたくさんあります。振動，回転などから，これからやる「音」に関しても波形にして記述することができます。

音を波形にして表すには，まずマイクなどを利用して音を録音します。マイクはいわば変換機で，これによって取り込むことができる音は，時間軸に従って変化する電圧，もしくは電流の変移と見ることができます。

この電圧などの変移を直接記録するものにアナログテープなどがあります。しかし，コンピュータで扱うには，デジタル化をしなくてはなりませんから，これをA/D変換機を通してメモリなどに保存します。これをサンプリング（標本化）といいます。

標本化は「とある間隔τ」で定期的に，アナログ値をデジタルに丸めて行うので，データは間隔τで離散的なものになり，値を丸めた分だけいい加減になります。

このτの間隔の逆数がサンプリングレートです。丸め後のデジタルデータがサンプリングビット数ですから，サンプリングレートが大きい，すなわちτが小さいほど高周波までサンプリングでき，ビット数が大きいほど，小さい波形の動きを読み取ることができます。

数学的には$\tau=0$で，ビット数が無限大なら，すべての波形がデジタル化できることになります。

標本化に関する詳しい話はまたの機会としておいといて，離散化された標本データの話を進めましょう。

その昔，Sampling PRO-68Kというソフトがありました。波形を直接マウスでエディットできる代物だったので，これさえあれば「原理的には」どんな音でも作れるというツールでした。が，これで，ちゃんとした音を作れたことがある人って，いったいどのくらいいるのでしょうか？　普通の人間には，頭にイメージする音を波形エディタで表記することなんてとてもできません。

シンセサイザと呼ばれるMIDI楽器はたくさんありますが，YAMAHAのFM音源からRolandのLA音源，KORGのAI²音源まで，ほとんどすべてのデジタル音源は最終段でD/Aコンバータを通ります。つまりPCMになるわけです。そのPCMを作る方法が，楽器屋さんのノウハウになるわけです。砕いていえば，わかりにくいPCM波形を，どうすれば感覚的に操作できるか？というところがシンセサイザの勝負どころともいえます。

この感覚を助ける作業に，大いに数学が役立ってくるともいえるわけです。

波形e(t)

とある波形を$e(t)$と表すことにしましょう。tはいうまでもなく時間です。この波形$e(t)$のパワーは，

$$P(t)=\{e(t)\}^2$$

と表し，この波形$e(t)$の$t_1\sim t_2$の区間のエネルギーを，

$$W(t)=\int_{t1}^{t2}P(t)dt$$

と表すことにします。波形のパワーは常に「正」の数で表され，エネルギーはその時間の積分なので，なにか区間指定が必要になります。

ここで，波形をいくつかに分けることにします。まず，

$$W(t)=\int_{-\infty}^{\infty}P(t)dt$$

とします。

これが，有限なものと無限なものです。

有限なものは，いつかは$P(t)=0$になる波形ですからこれは孤立的波形です。

無限のものは，永遠にある音であり，これは継続的波形です。

孤立的な波形は「有限」ですから，コンピュータでも扱うことができますが，真に無限のものをコンピュータで扱うことは不可能です。

そこで，継続的波形を分類することにします。まずは周期的な波形です。代表的なものは，$e(t)=\sin t$などでしょう。周期的な波形の特徴は，$e(t\pm NT_1)=e(t)$と表せることです。このなかでNは任意の整数，T_1はこの波形の周期です。これの逆数を1秒間の繰り返し回数として，基本周波数fといいます。

この周期的波形は無限に続くものですが，1周期ごとに区切ることができるため，コンピュータで扱うことは容易です。が，自然界において，完全周期的な波形はそう簡単には存在しません。

次に概周期的波形です。これは見かけ上は非周期的な波形ですが，周期の異なるいくつかの周期波形を加えただけのものです。これもコンピュータで解析できます。シンセサイザで扱う音の大半はこの概周期的波形ともいえます。

そして，非周期的波形です。これが本来なら自然界にいちばん多くあるものですが，残念ながら，コンピュータで扱うのは不可能に近くなってしまいます。

このほか，過渡的波形というものもあります。これはある一定の状態から具合が変わった別の一定状態までの「過渡状態」の波形です。過渡状態部分以外は一定の状態なので，これをコンピュータで解析することは容易です。

これらの中でフーリエ変換で解析するのは，孤立的波形，周期的波形，概周期的波形の3つです。参考までに，過渡的波形などはラプラス変換などを利用して解析します。

フーリエ変換

「○○変換」という言葉がありますが，これは簡単にいえば，座標系の変換です。たとえば高校の数学で，桁があまりにも大きなものなどを一度logにしてから計算をし，最後に指数を使って戻したりしませんでしたか？　つまり，解析や計算が難しいものを，一度なにかに変換して扱いやすい状態にし，加工してから戻す処理のことです。当然ながら，一度変換してからそのまま逆変換をかけた場合に同じものにならなければ変換とは呼びません。

音声のデータのx軸は時間軸で誰にでもわかることですが，y軸は電圧なので，実際の音にどのように影響するのかはさっぱりわかりません。さしずめ，わかるのは波形のパワーが人間の感じる音量に近いという程度で，音色，音の高さなどは，まるでわからないといっても過言ではないでしょう。

フーリエ変換というのは，この時間軸と電圧軸を周波数軸とその強さに置き換えるものです。とある音の離散的なデータを離散的フーリエ変換にかけると，その音に含まれる周波数ごとの強さがわかるということなのです。

こうして，フーリエ変換をかけると，その音が周波数ごとで，どのぐらいの強さになのかわかるので，低域フィルタでも高域フィルタでも好きなだけかけることができ，そして逆フーリエ変換をすることによって加工したあとにPCMデータに戻すということをするわけです。

結果的にはこういうことなのですが，原理はなかなか難しいものです。

最初に，波形を孤立的なもの，継続的だが周期的なもの，概周期的なものなどに分けました。フーリエ級数を手っとり早く説明できるのは，このうち周期的な波形のものです。

周期的な波形には，

$$e(t)=D_0+\sum_{t=1}^{\infty}[A_n\sin\{n(2\pi/T)t\}+B_n\cos\{n(2\pi/T)t\}]$$

ただし，

$$0<t<T,\ T=\frac{1}{f},\ 2\pi f=\omega$$

と，すべてがこの式に置き換えることができるという法則が成り立ちます。

この場合，Tはこの波形の周期で，fは周波数です（実際聞こえる「音」の周波数ではなく，周期を持った波形の周波数です）。

この波形e(t)を区間０～T，すなわちひとつの周期時間内における，フーリエ級数の展開表示を行うには，Persevelの完全条件式，

$$P=\frac{1}{T}\int_0^T\{e(t)\}^2dt=D_{0'}+\sum_{n=1}^{\infty}(A_{n^2}+B_{n^2})/2$$

から，このフーリエ係数は，

$$D_0=\frac{1}{T}\int_0^T e(t)dt$$

$$A_n=\frac{2}{T}\int_0^T e(t)\cdot\sin\{n(2\pi/T)t\}dt$$

$$B_n=\frac{2}{T}\int_0^T e(t)\cdot\cos\{n(2\pi/T)t\}dt$$

と表すことができます。

D_0は直流成分，A_nはsin成分，B_nはcos成分です。

このように，周期のある波形の場合はあっさりと手っとり早く説明することができます。Persevelの完全条件式は，これが成り立たないと波形解析は成り立たないというものです。

ところで，孤立的波形にPersevelの完全条件式を成り立たせる場合にはどうすればよいのでしょうか？　フーリエ変換は，あくまでも周期Tがわかっていることが前提です。孤立波形には周期はありませんが，周期を∞と考えると，範囲０～Tは－∞～＋∞ということになります。

T＝>∞に近づくわけですから，当然fは０に近づきます。まあ，ここでlimをつかった「うんぬん」の説明がありまして（詳しくは参考文献を参照），結果的に孤立的な波形の場合は，

$$e(t)=\frac{a}{2\pi}\int_{-\infty}^{+\infty}\{a(\omega)\cdot\sin\omega t+b(\omega)\cdot\cos\omega t\}d\omega$$

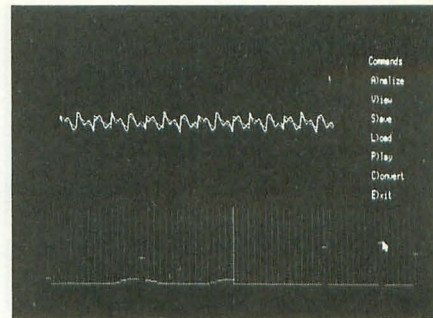
と，フーリエ積分され，

$$a(\omega)=2\int_{-\tau 2}^{+\tau 2}e(t)\cdot\sin\{\omega t\}dt$$

$$d(\omega)=2\int_{-\tau 2}^{+\tau 2}e(t)\cdot\cos\{\omega t\}dt$$

と逆フーリエ変換することができるというのが原理です。

孤立波形までわかったら，あとは概周期波形です。コンピュータを使った音源でもっとも多いのは，この概周期波形といえます。

概周期波形と周期波形の違いを砕いていってみますと，周期波形は「波形の周期＝音の周期」になりますが，概周期波形は「波形の周期＝音の周期にならない」ことです。これは別に，たいそうな理屈があるわけではありませんが，えてしてそうだと思っています。音の周期の逆数＝音の周波数，すなわち音の高さですから，ここまでいえば，もう簡単にわかりますよね？

FFTで再現した波形

概周期波形をFFTにかける場合，周期がわかりません。厳密に周期を求めていく方法がないことはないのですが，これはいつまでたっても終わりそうもない計算になります。

そこで，今回，眉唾ですが，周期t～t＋τの「τ」をいい加減に区切って決めることにします。

本来フーリエ変換というのは無限項までの計算ですから，概周期波形の場合正しい答を出すためには，無限に計算を続けなくてはなりません。フーリエ変換によってできる高調波は，大きくなるほど高周波を表すため，人間が聞く「音」にはあまり意味がなくなってきます。

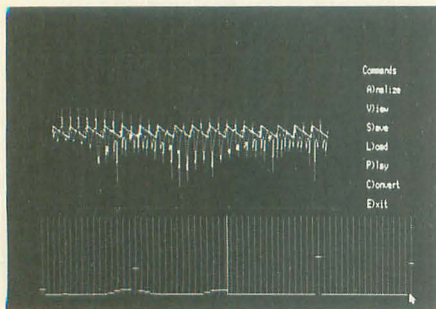
したがって，無限項まで計算せず，適当なところでやめておくのが筋になります。

周期τを仮に，2^nと固定した場合，フーリエ変換はクーリー・チューキーのアルゴリズムを利用した，いわゆるファーストフーリエ変換(FFT)が利用できます。これによって計算量が激減できるわけです。このFFTの詳しい説明は，野畠氏の記事に任せることにしましょう。

結果，τの逆数が基調波になるため，実際の音の基調波と関係はなくなりますが，どのあたりの高調波にどのぐらいの成分があるか分析できるので，音の高さなどを知ることができることになります。

プログラムについて

もともとのプログラムは，本誌1991年12月号の石上氏の記事，「冬の夜長のスペクトル解析」で掲載されたプログラムです。もともとxgcc＋xclibだったので，XGCC＋libcでコンパイル出来るように改変したかったのですが，時間不足ということもあって，XCのbaslib.l, zmusic.lを利用しています。

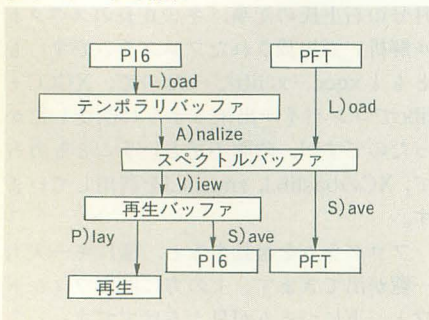
プログラムを起動すると，横にキー入力一覧が出てきます。上の方に波形ウィンドウと，下につまみが見えるはずです。

高調波の分布を解析

波形をロード(l)するときに，拡張子を判別します。*.PFTという拡張子以外は，すべてP16と同じ形式，符号付き16bit Bigendianと考えます。16bitPCMをロードすると，一度テンポラリバッファにロードされます。ロードされると，水色で波形ウインドウに波形を表示するはずです。長さが2048バイト以下なら，そのあとは0で埋められます。

そして，波形アナライズコマンド(a)により，フーリエ変換されてスペクトルバッファに転送されます。スペクトルバッファの内容は，第512次高調波まで波形表示ウインドウに表示されます。第32次高調波までは，下のつまみによって操作できます。左がcos成分，右がsin成分です。

ただ，32次高調波といっても，基調波の周波数が15Hz程度なので，32次高調波は500Hz弱です。したがって32次程度までいじってもフーリエ，逆フーリエの趣程度しかわからないでしょう。内部的には1024次高調波までサポートしているので，気が向いた方は遊んでみるのもよいでしょう。

この状態でSコマンドで拡張子をPFTにすると，スペクトルバッファの内容を書き出します。V)iewコマンドを実行すると，逆フーリエ変換をし，再生バッファにデータを復元します。Sコマンドでセーブするときに拡張子をPFT以外にすると，この再生バッファがセーブされます。

また，この再生バッファはPコマンドで再生することができます。素直にIOCSをコールしているので，PCM8を入れておくと，音がまともに鳴らないので注意してください。

これらの相関関係は表1に表しました。

機能は最低限のものしかありませんが，PFTファイルをもっと操作できるものを作ると，広域をカットしたりすることができるはずです。

あと，コンバートモードで，P16→PFT，PFT→P16に相互変換が可能です。これはあとで取ってつけた機能なので，1024ポイント以上のデータの場合，データを1024ポイントに分けて，フーリエ変換して保存します。

表1　データの流れ

PFTデータに関して

PFTデータは，表2に表すようなフォーマットになっています。データをSampPoint＝τで区切って，FFTにかけて叩き出された結果，すなわち高調波を表す構造体の内容をそのまま保存したものです。DataSizeは，元ファイルの大きさを保存するところですね。

今回，dataはdoubleの複素数なので，変換後はデータが肥大します。ただし，もともと8MHz÷512＝15625HzのPCMですから，基調波は，15625（1秒間のデータの数)÷1024（FFTにかけるポイント数)＝15.26Hzになるわけです。

データは1024次高調波まで含めるので，結果的に15625Hzまでしか考えていませんが，分析されたFFTの結果を見ると，意味のなさそうなスペクトルもいくつもあるはずです。その部分を勝手に0にしてしまっても，ほとんど音は変わらなかったりするわけです。

高調波に相当する部分を適当にきってしまうと，音はローパスフィルタを通したような音になります。また，基調波に近い低周波を切ってしまうと，音はハイパスフィルタを取ったような音にもなります。

まあ，これは遊びですが。

たとえば，このdouble複素数で表されているスペクトルデータを強引に2バイトの固定小数点に直したり，意味のなさそうな部分を強引に0にするなり，加工することによって2次圧縮が期待できるPCMデータなることもあります。

このあたりは調べてみると，面白いかもしれません。

無謀にも音を作ってみる

では，このプログラムの逆FFT機能を使って，無謀にも音を作ってみたいと思います。

中心線よりも右側がsin成分，左側がcos成分です。いちばん左の線は15Hzですから，人間の耳にはほとんど識別できない音です。たとえば，440HzのSIN波を作りたい場合，440÷15≒30ですから，左から30個目が，約440Hzということになります。

30個目の音の場合，2倍音は60個目になります。残念ながらこのプログラムでは32までしか使えないので，これはひとつの不満点です。もっとも，横に1024個のつまみがあると，それはそれで困るでしょうから，やっぱりなにかしらユーザーインタフェイスを考えなくてはなりませんけど，このことについては今後の課題ということにしましょう。

ということで，1オクターブ下の15個目を基準音(O3A)ということにしたいところですが，このあとの倍音処理にめんどうなことが起きるため，16個目ということにします。

どうせ，このプログラムでは，高調波のエディットはできないのですから，低めの音に挑戦……ということで，船の警笛です。16個目の倍音成分は，2倍音で32，1.5倍音で24あたりです。整数を立たせるほど，SIN波からあまり変わらない音になりますが，SIN波は基本ということで，32個目，1/2倍音の8個目，1/4倍音の4個目，1/8倍音の2個目を適当に立たせます。波形を見ても音なんてわかりっこないから，つまみをいじったらV)iew，P)layの繰り返しです。

表2　PFTファイルのフォーマットについて

```
0x0000  char   HEADER[4]  ="PFTF";ヘッダ
0x0004  int    SampPoint;          標本化の数
0x0008  int    DataSize;           元データサイズ
0x000c  int    Multiple;           倍周音の数
0x0010  char   RESERVE[32];        ヘッダ予約領域
0x0030 { double COS I
         double SIN I
              :              :     データ本体
       double  COS Multiple
          double SIN Multiple }
     以下 ((DataSize+SampPoint-1)/SampPoint)-1個連続
```

大きく倍，倍につまみをいじったら，その周りは，音に揺らぎを与えるため，適当に立たせることにします。刺型ですね。これが，FM音源でいうデチューンと同じ効果を作ります。

あとは，適当に……といっても，実はFFTで音を作るノウハウもたいしてないので，ほんとに適当になってしまいます。1/4倍音の3倍音などをうまいところ，イメージして，それで終わりです。

今回cos成分は使いませんでしたが，cos成分をうまく使うと，結構味が出て楽しい音が作れます。もっとも，下手に使うとたちまち，発振しはじめますけど。

おわりに

FFTでスペクトル解析して，逆FFTで元に戻せば，同じ音が鳴るはずです。ほとんど同じ結果が出ますが，長い概周期的な音をコンバートモードで変換すると，ときおり，ちょっとしたノイズが乗ります。

これは，本文中眉唾である部分，すなわち勝手にτを決める部分に大きく影響しています。勝手にτを決めたとしても，たとえば，ガウシャンパルスと畳み込み積分しながら行ったりすれば，ノイズの原因は少なくなっていくはずです。まあ，そうすればそうするほど，どんどん，実用的なスピードではなくなっていきますが……。

時間不足で実験的なものしかできませんでしたが，プログラムを見直せば，結構面白いものもできるかと思います。

あとは，各自で遊んでみてください。

参考文献

中野道夫，パルス回路入門，オーム社

リスト

```
  1: /*
  2:
  3: コンパイルの方法について
  4:
  5:   元ソースである石上氏のものは、XCライブラリによって記述されています。GCC
  6: によってコンパイル出来るように改変しましたが、XC ライブラリを各所で利用し
  7: ているため、コンパイルするには何等かの形でXCライブラリを購入しなくてはなり
  8: ません。また、P16->PCM への変換にZMUSICライブラリを利用しているため、
  9: ZMUSICシステムを買うなり、ダウンロードしなくてはなりません。
 10:   C言語のインストールは、各自で行ってください。参考までに。
 11:
 12:  $ set GCC_OPTION= LFIAMWOE+
 13:  $ set MARIKO=AB
 14:
 15:  $ gcc main.c -O -Wall -m68040 -fomit-frame-pointer -fstrength-reduce
 16:   -fforce-mem -fforce-addr -fcombine-regs -finline-functions -ldos
 17:   -liocs baslib.l floatdrv.l zmusic.l
 18:
 19: -----------------------------------------------------------------
 20:
 21:  $Id: main.c,v 1.7 1995/01/29 05:02:09 Kohju Exp $
 22:  $Log: main.c,v $
 23:  * Revision 1.7  1995/01/29  05:02:09  Kohju
 24:  * ・ConvertModeでは可変長が扱える用にした。
 25:  *
 26:  * Revision 1.6  1995/01/28  10:27:15  Kohju
 27:  * ・convert mode の追加。
 28:  * (ただし、まだBUFFSIZE固定長)
 29:  *
 30:  * Revision 1.5  1995/01/28  09:58:04  Kohju
 31:  * ・pft形式のローダ、セーバが付いた。
 32:  *
 33:  * Revision 1.4  1995/01/28  08:26:41  Kohju
 34:  * ・再生のバグを修復
 35:  *
 36:  * Revision 1.3  1995/01/28  07:35:57  Kohju
 37:  * ・音声を鳴らすコマンドを追加した。
 38:  *
 39:  * Revision 1.2  1995/01/28  06:26:54  Kohju
 40:  * ・ワーニングが出ないGCC&LIBC用のソースになった。
 41:  *
 42:  * Revision 1.1  1995/01/27  10:34:22  Kohju
 43:  * Initial revision
 44:  * Authorが私になっているけど、Rev 1.1は石上氏の原生ソース
 45:
 46: */
 47:
 48: #define                 __DIRECT_FPU__
 49: // 68030+68882/1を搭載した機種(含むXellent030,040turbo)
 50: //#define  __DIRECT_IOFPU__
 51: // CompactXVI未満で、FPU68882/1をつけた機種(CPUは68000)
 52: //#define  __DIRECT_FLOAT__
 53: // 上記2つ以外。
 54: // いずれかのコメントを外すと、マシンにあわせて最適化されます。
 55:
 56: #include   <stdio.h>
 57: #include   <stdlib.h>
 58: #include   <sys¥dos.h>
 59: #include   <sys¥iocs.h>
 60: #include   <sys¥stat.h>
 61: #include   <fcntl.h>
 62: #include   <io.h>
 63: #include   <unistd.h>
 64: #include   <math.h>
 65: #include   <ctype.h>
 66: #include   <string.h>
 67: #include   <zmusic.h>
 68:
 69: typedef struct cmplx {      /* 複素数型 */
 70:     float   Re;
 71:     float   Im;
 72: } CMPLX;
 73:
 74: #define     NASI            0x4e415349      /* 'NASI' */
 75: #define     BUFFSIZE        1024            // バッファサイズとfftの第n次高調波
 76: #define     PI              M_PI            // PI
 77: #define     LOOPTIMES       32              // 再生モードでの倍算ループ回数
 78: //#define   __P16_L_SAVE__                  // 定義されると実際に再生されるデータを
 79:                                             // セーブします。
 80:
 81: //          プロシージャ、関数宣言
 82:
 83: void        clrWin(int );
 84: void        mslDown(int , int );
 85: void        msrDown(int ,int );
 86: void        initScreen( void );
 87: void        drawVr(int );
 88: void        erasVr( int );
 89: void        drawGraph(short *, int );
 90: int                 loadData( void );
 91: int                 saveData( void );
 92: void        analize( void );
 93: void        view( int );
 94: void        PlayP16( void );
 95: void        fft(const CMPLX *, CMPLX *, int ,int );
 96: int                 getlastext( char *, char *_ext,const char *);
 97: void        Convert( void );
 98:
 99: //          baslib内、外部関数宣言。
100:
101: //          初期化関数
102: void        console(int, int, int);
103: void        screen(int, int, int, int);
104: void        b_input(char *, int,...);
105: //          graphic関数
106: void        box(int, int, int, int, int, int);
107: void        line(int, int, int, int, int, int);
108: void        fill(int, int, int, int, int);
109: void        wipe(void);
110: //          mouse制御関数
111: int                 mouse(int);
112: int                 msarea(int, int, int, int);
113: int                 mspos(int *, int *);
114: int                 msstat(int *, int *, int *, int *);
115: char        *b_inkey0(char *);
116:
117:     CMPLX   c[BUFFSIZE+1],
118:                     y[BUFFSIZE+1],
119:                     z[BUFFSIZE+1];
120:
121:     double  s[64+1];
122:     CMPLX   LevelM[BUFFSIZE+1];
123:     short   data[BUFFSIZE+1];
124:     short   moto[BUFFSIZE+1];
125:
126:     char    fileName[40];                   // ファイル名
127:     char    ConvFN[40];                     // ファイル名(コンバートモードで利用)
128:     int             fn;                     // ファイルハンドル
129:     int             ConvertMode=0;          // ConvertMode ON(=1)/OFF(=0)
130:     int             NoEmpty=0;              // ConvertModeでFileがEmptyでなければ1
131:
132: void        main( int argc , char **argv)
133: {
134:     char    strtmp0[258];
135:     char    key[1];
136:     int             i;
137:     int             ms_x,ms_y,ms_bl,ms_br;
138:
139:     console(0,32,0);
140:     initScreen();
141:     for (i=0;i<=512;i++){
142:             data[i]=0;
143:     }
144:
145:     box(30,30,30+512,30+256,8,NASI);
146:     line(30,30+128,30+512,30+128,8,NASI);
147:     mouse(4);
148:     mouse(1);
149:     msarea(6,300,699,428);
150:     _iocs_b_curoff();
151:
152:     while (1) {
153:             msstat(&ms_x,&ms_y,&ms_bl,&ms_br);
154:             mspos(&ms_x,&ms_y);
155:             if(ms_bl == -1) mslDown(ms_x, ms_y);
156:             if(ms_br == -1) msrDown(ms_x, ms_y);
157:
158:             _iocs_b_locate(77,19);
159:             strncpy(key, b_inkey0(strtmp0), sizeof(key));
160:                                     // リアルタイム一文字入力
161:
162:             switch(toupper(*key)) {
163:             case 'A':
164:                     analize();
165:                     break;
166:             case 'E':
167:                     mouse(0);
168:                     wipe();
```

```
169:                            _iocs_b_locate(0,31);
170:                            exit(0);
171:                            break;
172:            case 'V':
173:                            view( 0 );
174:                            break;
175:            case 'L':
176:                            loadData();
177:                            break;
178:            case 'S':
179:                            saveData();
180:                            break;
181:            case 'P':
182:                            PlayP16();
183:                            break;
184:            case 'C':
185:                            Convert();
186:                            ConvertMode=0;
187:                            break;
188:                    }
189:        }
190: }
191:
192: void        clrWin(int line)
193: {
194:    int     cnt;
195:
196:    _iocs_b_locate(0,28);
197:    while(line--) {
198:            cnt = 95;
199:            while(cnt--) putchar(' ');
200:            putchar('¥n');
201:    }
202: }
203:
204: /*
205:    マウスの左ボタンが押された
206: */
207: void        mslDown(int x, int y)
208: {
209:    int     ch;
210:
211:    ch=(x - 3) / 11;
212:    erasVr(ch);
213:    if(s[ch] >= 0)  s[ch]= pow((double)10.0,(428-y) / 30.0);
214:    else                    s[ch]=-pow((double)10.0,(428-y) / 30.0);
215:    drawVr(ch);
216: }
217:
218: /*
219: **  マウスの右ボタンが押された
220: */
221: void        msrDown(int x,int y) {
222:    int     ch;
223:    int     tmp;
224:
225:    ch=(x - 3) / 11;
226:    _iocs_b_locate(0,28);
227:    if ( x < 353 ) {
228:            printf("ＳＩＮ関数の%d番目の係数の値は%4.2fです¥n",
229:                    ch+1,s[ch]);
230:            } else {
231:            printf("ＣＯＳ関数の%d番目の係数の値は%4.2fです¥n",
232:                    ch-31,s[ch]);
233:    }
234:    b_input("新しい値を入れて下さい－>",0x204,&tmp,-1);
235:    erasVr(ch);
236:    s[ch]=tmp % 0x8000;
237:    drawVr(ch);
238:    clrWin(2);
239: }
240:
241: /*
242: **  画面の初期化
243: */
244: void        initScreen( void )
245: {
246:    int     i;
247:
248:    screen(2,0,1,1);
249:    for (i=0;i<=31;i++){
250:            line(6+i*11,300,6+i*11,428,9,NASI);
251:            line(358+i*11,300,358+i*11,428,9,NASI);
252:    }
253:    line(351,300,351,428,5,NASI);
254:    _iocs_b_locate(76,3);
255:    puts("Commands ");
256:    _iocs_b_locate(77,5);
257:    puts("A)nalize");
258:    _iocs_b_locate(77,7);
259:    puts("V)iew ");
260:    _iocs_b_locate(77,9);
261:    puts("S)ave ");
262:    _iocs_b_locate(77,11);
263:    puts("L)oad ");
264:    _iocs_b_locate(77,13);
265:    puts("P)lay ");
266:    _iocs_b_locate(77,15);
267:    puts("C)onvert");
268:    _iocs_b_locate(77,17);
269:    puts("E)xit ");
270:
271:    for (i=0;i<=63;i++){    /*ボリュームを描く*/
272:            s[i]=0;
273:            drawVr(i);
274:    }
275: }
276:
277: /*
278: **  ボリュームを描く
279: */
280: void        drawVr(int ch)
281: {
282:    int     x,y;
283:
284:    x = ch*11;
285:    if(s[ch] == 0)
286:            y = 428;
287:    else
288:            y = 428-log10(fabs(s[ch]))*30;
289:
290:    if( y < 300) y = 300;
291:    if( y > 428) y = 428;
292:    box(x,y,x+11,y,9,NASI);
293: }
294:
295: /*
296: **  ボリュームを消す
297: */
298: void        erasVr( int ch)
299: {
300:
301:    int     x,y;
302:
303:    x = ch*11;
304:    if(s[ch] == 0)
305:            y = 428;
306:    else
307:            y = 428-log10(fabs(s[ch]))*30;
308:
309:    if( y < 300) y = 300;
310:    if( y > 428) y = 428;
311:
312:    fill(x,y,x+11,y,0);
313:
314:    line(6+ch*11,300,6+ch*11,428,9,NASI);
315:    if ( ch==31 | ch==32 ) {
316:            line(351,300,351,428,5,NASI);
317:    }
318: }
319:
320: /*
321: **  画面に波形を描く
322: */
323: void        drawGraph(short *dat, int clr)
324: {
325:    int     oy,ny;
326:    int     i;
327:
328:    oy=158-dat[0]/32;
329:    for (i=1; i <= 512 ; i++) {                     // Here!
330:            ny = 158 - dat[i*BUFFSIZE/512] / 32;
331:            if(ny > 30+256) ny = 30+256;
332:            if(ny < 30)     ny = 30;
333:            line(i+29,oy,i+30,ny,clr,NASI);
334:            oy=ny;
335:    }
336:    box(30,30,30+512,30+256,8,NASI);
337:    line(30,30+128,30+512,30+128,8,NASI);
338: }
339:
340: /*
341:    ファイルより読み込む
342: */
343: int                 loadData( void )
344: {
345:
346:    int             i=0,j=0;
347:    int             flen=0;                 // ファイル長
348:    int             outhandle=0;
349:    char    limit_ext[128]; // 拡張子をひとつ除外されたファイル名
350:    char    ext[32];                        // 拡張子
351:    int             pft[12];
352:    int             SampPoint=0;
353:    int             DataSize=0;
354:    int             Multiple=0;
355:    int             bytes=0;
356:    int             stcmpe=0;
357:    int             Reserve[8]={0,0,0,0,0,0,0,0};
358:
359:    if(ConvertMode==0){
360:            _iocs_b_locate(0,28);
361:            b_input("File Name: ",sizeof(fileName),fileName,-1);
362:            clrWin(1);
363:            getlastext(ext ,limit_ext ,fileName);   // 拡張子の判別
364:            stcmpe=stricmp(ext,"pft");
365:    }
366:    else if(ConvertMode==1){                // P16->PFT
367:            strmfe(fileName,ConvFN,"p16");  // ファイル名の拡張子を無条件で
368:            stcmpe=1;                       // p16に変換する。
369:    }
370:    else if(ConvertMode==2){        // PFT->P16
371:            strmfe(fileName,ConvFN,"pft");  // ファイル名の拡張子を無条件で
372:            stcmpe=0;                       // pftに変換する。
373:    }
374:
375:    if(stcmpe==0){                          // pftフォーマットなら
376:            if((fn=open(fileName, O_BINARY | O_RDONLY)) < 0) {
377:                    _iocs_b_locate(0,28);
378:                    puts("ファイルがオープン出来ません");
379:                    return(-1);
380:            }
381:            for(i=0;i<=BUFFSIZE;i++) c[i].Re=0;
382:            for(i=0;i<=BUFFSIZE;i++) c[i].Im=0;
383:            if((int)(bytes=read(fn,(void *)pft,0x30)) < 0){
384:                    _iocs_b_locate(0,28);
385:                    puts(   "ファイルが読めません。");
386:                    close(fn);
387:                    return(-3);
388:            }
389:            if(pft[0]!='PFTF') {
390:                    _iocs_b_locate(0,28);
391:                    puts("PFTファイルではありません");
392:                    return(-4);
```

▶２月号の88ページの平勝久さんのコメントに「千葉電子」とあるのを「チバデンコ」と読んでしまった。恐るべし，満開製作所！　黒澤　典義(17)群馬県

```
393:            }
394:            SampPoint=pft[1];
395:            DataSize =flen=pft[2];
396:            Multiple =pft[3];
397:            _iocs_b_locate(0,28);
398:            printf( "¥t        標本化数:%d¥n"
399:                    "¥t復元データサイズ:%d¥n"
400:                    "¥t        高調波の数:%d",SampPoint,DataSize,Multiple);
401:            if((int)(bytes=read(fn,(void *)&c[1],Multiple*sizeof(CMPLX)))<0){
402:                    _iocs_b_locate(0,28);
403:                    puts("ファイルが読み込めません");
404:                    return(-1);
405:            } else if(ConvertMode==0){
406:                    close(fn);
407:                    view(1);
408:                    clrWin(3);
409:            } else {
410:                    strmfe(ConvFN,fileName,"p16");  // ファイル名の拡張子を無条件で
411:                    if((outhandle=open(ConvFN,O_CREAT|O_RDWR|O_BINARY|O_TRUNC
412:                                            ,S_IREAD|S_IWRITE))<0){
413:                            _iocs_b_locate(0,28);
414:                            puts("ファイルがオープン出来ません");
415:                            return(-1);
416:                    }
417:                    _iocs_b_locate(0,31);
418:                    j=DataSize/BUFFSIZE/2;
419:                    for(i=0;i<=j;i++) _iocs_b_putc('.');
420:                    _iocs_b_locate(0,31);
421:                    do {
422:                            view(1);
423:                            if(flen<BUFFSIZE)
424:                                    write(outhandle, (void *)&data[1], flen *
sizeof(short)) ;
425:                            else    write(outhandle, (void *)&data[1], BUFFSIZ
E * sizeof(short));
426:                            flen-=BUFFSIZE*sizeof(short);
427:                            _iocs_b_putc('o');
428:                            for(i=0;i<=BUFFSIZE;i++) c[i].Re=0;
429:                            for(i=0;i<=BUFFSIZE;i++) c[i].Im=0;
430:                    } while ((int)(bytes=read(fn,(void *)&c[1],Multiple*sizeof
(CMPLX)))>0);
431:                    close(fn);
432:            }
433:    }
434:    else{                                   // P16フォーマット。
435:            if((fn=open(fileName, O_BINARY | O_RDONLY)) < 0) {
436:                    _iocs_b_locate(0,28);
437:                    puts("ファイルがオープン出来ません");
438:                    return(-1);
439:            }
440:
441:            if((bytes=read(fn, (void *)&data[1], BUFFSIZE * sizeof(short)))<0){
442:                    return(-1);
443:            } else if(ConvertMode==0) {
444:                    for(i=bytes; i<= BUFFSIZE; i++)data[i]=0;
445:            } else {
446:                    strmfe(ConvFN,fileName,"pft");  // ファイル名の拡張子を無条件で
447:                    if((outhandle=open(ConvFN,O_CREAT|O_RDWR|O_BINARY|O_TRUNC
448:                                            ,S_IREAD|S_IWRITE))<0){
449:                            _iocs_b_locate(0,28);
450:                            puts("ファイルがオープン出来ません");
451:                            return(-1);
452:                    }
453:                    write(outhandle,(void *)"PFTF",4);
454:                    SampPoint=BUFFSIZE;                     // 標本化した数
455:                    write(outhandle,(void *)&SampPoint,4);
456:                    flen=filelength(fn);                    // 元データのサイズ
457:                    DataSize=flen;
458:                    write(outhandle,(void *)&DataSize,4);
459:                    Multiple=BUFFSIZE;                      // 倍音データの数
460:                    write(outhandle,(void *)&Multiple,4);
461:                    write(outhandle,(void *)&Reserve,32);
462:                    _iocs_b_locate(0,29);
463:                    j=flen/BUFFSIZE/2;
464:                    for(i=0;i<=j;i++) _iocs_b_putc('.');
465:                    _iocs_b_locate(0,29);
466:                    do{
467:                            fill(31,31,30+511,30+255,0);
468:                            drawGraph(data, 11);
469:                            for(i=0; i<= BUFFSIZE; i++) moto[i]=data[i];
470:                            analize();
471:                            if(write(outhandle,(void *)&c[1],BUFFSIZE*sizeof(C
MPLX))<0){
472:                                    _iocs_b_locate(0,28);
473:                                    puts("ファイルが書き込めません¥n");
474:                                    return(-1);
475:                            }
476:                            _iocs_b_putc('o');
477:                            for(i=0; i<= BUFFSIZE; i++)data[i]=0;
478:                    } while((bytes=read(fn,(void *)&data[1], BUFFSIZE*sizeof(s
hort)))>0);
479:            }
480:            close(fn);
481:
482:            fill(31,31,30+511,30+255,0);
483:            drawGraph(data, 11);
484:
485:            for(i=0; i<= BUFFSIZE; i++)
486:                    moto[i]=data[i];
487:    }
488:    clrWin(3);
489:    return(0);
490: }
491:
492: /*
493:    ファイルに保存する
494: */
495:
496: int     saveData( void)
497: {
498:    char    limit_ext[128]; // 拡張子をひとつ除外されたファイル名
499:    char    ext[32];                        // 拡張子
```

```
500:    int             SampPoint=BUFFSIZE;
501:    int             DataSize=BUFFSIZE;
502:    int             Multiple=BUFFSIZE;
503:    int             Reserve[8]={0,0,0,0,0,0,0,0};
504:
505:    _iocs_b_locate(0,28);
506:    b_input("File Name: ",sizeof(fileName),fileName,-1);
507:    clrWin(1);
508:
509:    getlastext(ext ,limit_ext ,fileName);   // 拡張子の判別
510:
511:    if(stricmp(ext,"pft")==0){                      // pftフォーマットなら
512:            if((fn=open(fileName,O_CREAT|O_RDWR|O_BINARY|O_TRUNC
513:                                    ,S_IREAD|S_IWRITE))<0){
514:                    _iocs_b_locate(0,28);
515:                    puts("ファイルがオープン出来ません");
516:                    return(-1);
517:            }
518:            write(fn,(void *)"PFTF",4);
519:            write(fn,(void *)&SampPoint,4);
520:            write(fn,(void *)&DataSize,4);
521:            write(fn,(void *)&Multiple,4);
522:            write(fn,(void *)&Reserve,32);
523:            if(write(fn,(void *)&c[1],BUFFSIZE*sizeof(CMPLX))<0){
524:                    _iocs_b_locate(0,28);
525:                    puts("ファイルが書き込めません¥n");
526:                    return(-1);
527:            }
528:            close(fn);
529:    }
530:    else {                                          // それ以外は16bit bigendian
531:            if((fn=open(fileName, O_CREAT|O_RDWR|O_BINARY|O_TRUNC
532:                                    ,S_IREAD|S_IWRITE))<0) {
533:                    _iocs_b_locate(0,28);
534:                    puts("ファイルがオープン出来ません");
535:                    return(-1);
536:            }
537:            write(fn, (void *)&data[1], BUFFSIZE * sizeof(short));
538:            close(fn);
539:    }
540:    return(0);
541: }
542:
543: void    PlayP16( void )
544: {
545:    char    *pcm;
546:    short   *p16;
547:    int             inc,inc2;
548: #ifdef          __P16_L_SAVE__
549:    int             handle=0;
550: #endif
551:    _iocs_b_locate(0,28);
552:    puts("Converting...");
553:
554:    if((p16=malloc(BUFFSIZE*LOOPTIMES*sizeof(short))) == NULL ){
555:            _iocs_b_locate(0,28);
556:            puts("メモリが足りません。");
557:            return;
558:    }
559:
560:    if((pcm=malloc(BUFFSIZE*LOOPTIMES/2)) == NULL ){
561:            _iocs_b_locate(0,28);
562:            puts("メモリが足りません。");
563:    }
564:
565:    inc=0;
566:    while(inc<BUFFSIZE*LOOPTIMES){
567:            for(inc2=1;inc2<=BUFFSIZE;inc2++,inc++){
568:                    p16[inc]=data[inc2];
569:            }
570:    }
571:
572: #ifdef                  __P16_L_SAVE__
573:
574:    if((handle=open("foo.p16",O_CREAT|O_RDWR|O_BINARY|O_TRUNC
575:                                    ,S_IREAD|S_IWRITE))<0){
576:            printf("ファイルがオープン出来ません¥n");
577:            exit(-1);
578:    }
579:    if(write(handle,(void *)p16,BUFFSIZE*LOOPTIMES*sizeof(short))<0){
580:            printf("ファイルが書き込めません¥n");
581:            exit(-1);
582:    }
583:    close(handle);
584:
585: #endif
586:
587:    pcm_to_adpcm( p16,BUFFSIZE*LOOPTIMES*sizeof(short),pcm);
588:
589:    free(p16);
590:
591:    _iocs_b_locate(0,28);
592:    puts("再生中。SPACEを押すまでし続けます");
593:
594:    while((_iocs_bitsns(6) & 0x20)!=0x20 ){
595:            while(_iocs_adpcmsns()!=0)
596:                    if((_iocs_bitsns(6) & 0x20)==0x20) break;
597:            _iocs_adpcmout(pcm,0x403,BUFFSIZE*LOOPTIMES/2);
598:    }
599:    free(pcm);
600:
601:    clrWin(1);
602: }
603:
604: /*
605: **  波形を解析する
606: */
607: void    analize( void )
608: {
609:    int     i,Voly,oy;
610:    struct  _psetptr        PointP;
611:
```

▶缶コーヒーでいちばん美味しかったといわれる，コカコーラの緑ラベルのジョージアにあえる日はもう，ないのか……。　　今井　佑(17)東京都

```
612:    if(ConvertMode==0){
613:            _iocs_b_locate(0,28);
614:            puts("Now Calculating ...(Analize)");
615:    }
616:
617:    for(i=1; i <= BUFFSIZE; i++) {
618:            y[i].Re = moto[i+1];
619:            y[i].Im = 0;
620:    }
621:
622:    fft(y,c,BUFFSIZE,-1);
623:
624:    oy=158;
625:    for(i=1; i <= 512; i++) {
626:            LevelM[i].Re=c[i].Re;
627:            if(LevelM[i].Re == 0) Voly = 428;
628:            else Voly = 158-log10(fabs(LevelM[i].Re))*30;
629:            if( Voly < 30    ) Voly = 30;
630:            if( Voly > 30+256) Voly = 30+256;
631:            line(i+29,oy,i+30,Voly,13,NASI);
632:            oy=Voly;
633:    }
634:    oy=158;
635:    for(i=1; i <= 512; i++) {
636:            LevelM[i].Re=c[i].Re;
637:            if(LevelM[i].Re == 0) Voly = 428;
638:            else Voly = 158-log10(fabs(LevelM[i].Re))*30;
639:            if( Voly < 30    ) Voly = 30;
640:            if( Voly > 30+256) Voly = 30+256;
641:            line(i+29,oy,i+30,Voly,14,NASI);
642:            oy=Voly;
643:    }
644:
645:    for(i=0; i<=31; i++) {
646:            erasVr(i);
647:            s[i] = c[i+ 1].Re;
648:            drawVr(i);
649:    }
650:    for(i=32; i<=63; i++) {
651:            erasVr(i);
652:            s[i] = c[i-30].Im;
653:            drawVr(i);
654:    }
655:
656:    if(ConvertMode==0) clrWin(1);
657: }
658:
659: /*
660: **  波形を合成する
661: */
662: void        view( int lf)
663: {
664:    int     i;
665:
666:    if(lf==1){
667:            for(i=0; i<=31; i++) {
668:                    erasVr(i);
669:                    s[i] = c[i+ 1].Re;
670:                    drawVr(i);
671:            }
672:            for(i=32; i<=63; i++) {
673:                    erasVr(i);
674:                    s[i] = c[i-30].Im;
675:                    drawVr(i);
676:            }
677:    }
678:
679:    if(ConvertMode==0){
680:            _iocs_b_locate(0,28);
681:            puts("Now Calculating ...(VIEW)");
682:    }
683:
684:    for(i=1; i <= BUFFSIZE; i++) {
685:            y[i].Re = data[i];
686:            y[i].Im = 0;
687:    }
688:
689:    if(lf!=1){
690:            for(i=0 ; i<=31; i++) c[i+ 1].Re = s[i];
691:            for(i=32; i<=63; i++) c[i-30].Im = s[i];
692:    }
693:
694:    fft(c, z, BUFFSIZE,1);
695:
696:    for(i=1; i <= BUFFSIZE; i++) {
697:            data[i] = z[i].Re;
698:    }
699:    data[BUFFSIZE]=data[BUFFSIZE-1];
700:
701:    fill(31,31,30+511,30+255,0);
702:    drawGraph(moto, 11);
703:    drawGraph(data, 13);
704:
705:    if(ConvertMode==0)      clrWin(1);
706: }
707:
708: /*
709: **FFT解析プログラム（参考文献1による）
710: ** n:データの総数
711: ** iw == -1:フーリエ変換
712: **          *y:標本値(src)            *c:フーリエ級数(dist)
713: ** iw == 1:逆フーリエ変換
714: **          *y:フーリエ級数(src) *c:出力(dist)
715: */
716: void        fft(const CMPLX *y, CMPLX *c, int n,int iw)
717: {
718:    CMPLX   str;
719:    float   theta;
720:    int     i,is,j,k,m,max;
721:
722:    for( i=1 ; i <= n; i++) {
723:            c[i].Re = y[i].Re;
724:            c[i].Im = y[i].Im;
725:    }
726:    for( i = j = 1 ; i <= n; i++) {
727:            if( i < j)  {   /* swap(&c[i],&c[j]) */
728:                    str.Re = c[j].Re; c[j].Re = c[i].Re; c[i].Re = str.Re;
729:                    str.Im = c[j].Im; c[j].Im = c[i].Im; c[i].Im = str.Im;
730:            }
731:            m = n / 2;
732:            while( j > m ) {
733:                    j = j - m;
734:                    m = m / 2;
735:                    if(m < 2) break;
736:            }
737:            j = j + m;
738:    }
739:    max = 1;
740:    while(max < n) {
741:            is = max * 2;
742:            for( k = 1; k <= max; k++) {
743:            theta = PI * iw * (k - 1) / max;
744:                    for( i = k; i <= n; i+= is) {
745:                            j = i + max;
746:                            /* str = c[j] * exp(theta) */
747:                            str.Re = c[j].Re * cos(theta) - c[j].Im * sin(the
ta);
748:                            str.Im = c[j].Re * sin(theta) + c[j].Im * cos(the
ta);
749:                            c[j].Re = c[i].Re - str.Re;
750:                            c[j].Im = c[i].Im - str.Im;
751:                            c[i].Re = c[i].Re + str.Re;
752:                            c[i].Im = c[i].Im + str.Im;
753:                    }
754:            }
755:            max = is;
756:    }
757:    if(iw != 1) {
758:            for(i=1; i<= n; i++) {
759:                    c[i].Re = c[i].Re / n;
760:                    c[i].Im = c[i].Im / n;
761:            }
762:    }
763: }
764:
765: int         getlastext( char *ext, char *limit_ext,const char *name)
766: /*
767: 機能:
768:    name が指すファイル名にから、
769:    最後に付加された拡張子を取り出し、ext に格納して返します。
770:    また、拡張子を除外したファイル名を、limit_ext に格納して返します。
771: 戻り値:
772:    最後のピリオドのポインタを返します。
773:    拡張子がまったくなかった場合は -1 を返します。
774: */
775:
776: {
777:    int     len=0,si=0,di=0,end=0,result=-1,i=0;
778:    si=len=(strlen(name)-1);
779:    do{
780:            if(name[si]=='.') {
781:                    result=si;
782:                    si++;
783:                    while(name[si]!=0){
784:                            ext[di]=name[si];
785:                            di++;
786:                            si++;
787:                    }
788:                    ext[di]=0;
789:                    for(i=0;i<result;i++) limit_ext[i]=name[i];
790:                    limit_ext[i]=0;
791:                    end=1;
792:            }
793:            if(name[si]=='/' ) {result=-1;end=1;}
794:            if(name[si]=='¥¥') {result=-1;end=1;}
795:            if(name[si]==':' ) {result=-1;end=1;}
796:            si--;
797:            if(si<=0) {result=-1;end=1;}
798:    } while(end!=1);
799:    return(result);
800: }
801:
802: void        Convert( void )
803: {
804:    char    temp[256];
805:    _iocs_b_locate(0,28);
806:    b_input("1.P16->PFT  or  2.PFT->P16 : ",sizeof(temp),temp,-1);
807:    ConvertMode=atoi(temp);
808:    clrWin(1);
809:
810:    _iocs_b_locate(0,28);
811:    b_input("File Name: ",sizeof(ConvFN),ConvFN,-1);
812:    clrWin(1);
813:
814:    _iocs_b_locate(0,28);
815:    if(ConvertMode==0) return;
816:    else if(ConvertMode==1){
817:            puts("1.P16->PFT");
818:            if(loadData()<0) return;
819:            analize();
820:            clrWin(2);
821:            return;
822:    }
823:    else if(ConvertMode==2){
824:            puts("2.PFT->P16");
825:            loadData();
826:            view(0);
827:            clrWin(4);
828:            return;
829:    }
830: }
```

音を加工する手法を見る

エフェクタ処理の実際

Nakano Shuichi 中野 修一

電子楽器の音はさまざまなフィルタを通して加工されている
単純な音も処理を重ねていくことで，ついには原形を留めないくらい様変わりする
ここではサウンドエフェクタの処理について考えてみよう

「SoundEffects」という言葉には2つの意味があります。ひとつは「効果音」としての意味，もうひとつは「音に対する特殊効果」としての意味です。前者はゲーム中での効果音などを思い浮かべてもらえばいいでしょうし，後者はエコーやディストーションなどの各種エフェクタの処理を想像してみればいいでしょう。

効果音というのはパソコンからのレスポンスのうち聴覚に訴えるものを表し，コミュニケーションの一形態として重要な意味を持ちます。たいていの効果音にはエフェクトがかかっていますから，どちらにしてもエフェクト処理に焦点をあてる必要がありそうです。

ここでは各種エフェクタというのがだいたいどのような原理で動いているのかを見てみましょう。

各種エフェクタの処理

音楽用のエフェクタには実にさまざまな種類があります。そういったものをまとめて取り上げ，プログラム化する際の方針を検討してみましょう。

●イコライザ

高音をカットして低音だけ通すものをローパスフィルタ，逆をハイパスフィルタ，特定の周波数帯だけ通すものをバンドパスフィルタといいます。こういったものをまとめて周波数帯ごとに出力量を決めようというのがイコライザです。

周波数帯ごとにスライダーボリュームなどで音量を調整する形式なのがグラフィックイコライザ（スペクトルアナライザと対になっていることが多い），特定の周波数帯を指定して，その帯域をどの程度の幅でどの程度ブーストまたはカットするかを決めるのがパラメトリックイコライザです。

・実装

PCMの信号レベルでは，どのあたりが高い音でどのあたりが低い音かというのはなかなかわかりません。こういったことを行うには，FFTなどで周波数成分に分解してデジタルフィルタ処理する必要があります。

ここでは逆FFTされたものの係数を直接いじって簡易イコライジング処理することを考えてみましょう。相当する周波数帯の係数をいじるだけなので，真面目にデジタルフィルタ設計をやっている人から怒られそうな気もしますが，真面目なフィルタリングに立ち入るとあまりに難しくなるので今回は踏み込むのはやめておきます。

今回瀧氏が作成したツールで周波数成分に展開されたファイルを操作してFFTにより音声を取り出します。ここでは1024ポイントの逆FFTをされたものの各係数をグライコ風にまとめていじっていくとよいでしょう。

ということで簡易イコライザを作ってみました(リスト1)。マウスで適当にいじって右ボタンで変換です。軸を対数でとるべきかとか，いろいろ悩みましたが，結局係数そのものを単純に増減させることにしました。この処理の意味するところの具体的な内容については突っ込まないように。

●ディストーション

以前の音楽特集でもディストーション処理について解説がありましたが，ディストーションの内容をひとことでいえば「特性の悪いアンプ」ということになります。音量を20%上げるつもりでいても，実は音の小さい部分では20%でも音が大きい部分では5%しか上がってなかった……といった場合を考えましょう。コンプレッサと似ていますが，アンプは信号レベルで随時作用していますから，先ほど周波数データに影響する云々といった事態がそのまま起こっているわけです。それによって波形が歪み，毛羽立った独特の音になるわけです。

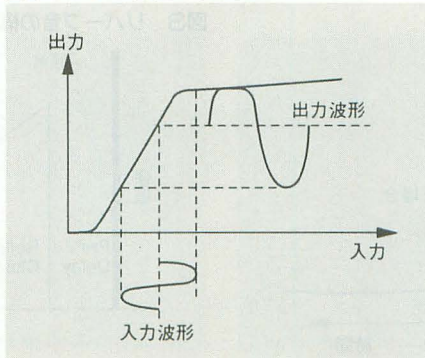



図1 アンプ特性による歪み

・実装

アンプ特性の変化する点（スレッショルドレベル）と変化率を決めます。スレッショルド以前の部分は1：1のフラットな特性のアンプとして，スレッショルド以降はどの程度の増幅率（減衰率か？）になるかを決めます。

リスト1

```
 10 int p(64),head(11)
 20 int a,b,c,d,e,f,g,h,x,y,r,l,siz,loop
 30 float cs(2047),w
 40 screen 1,3,1,1
 50 fill(0,0,511,511,6353)
 60 fill(245,11,500,137,35321)
 70 line(246,74,500,74,45472)
 80 for i=0 to 63
 90   fill(246+i*4,74,246+i*4+2,136,9999):p(i)=0
100 next
110 mouse(1):mouse(4)
120 repeat
130 msstat(x,y,l,r)
140 if l<>0 then {
150   mspos(x,y)
160   if point(x,y)<6354 then continue
170   a=(x-245)/4
180   b=74-y
190   if p(a)<b then fill(246+a*4,74-p(a),246+a*
4+2,74-b,9999)
200   if p(a)>b then fill(246+a*4,74-p(a),246+a*
4+2,74-b,35321)
210 locate 10,10:print b
220   p(a)=b
230 }
240 if r=-1 then equalize()
250 until r=-1
260 end
270 func equalize()
280 f=fopen("test.pft","r")
290 f2=fopen("test2.pft","c")
300 fread(head,12,f):fwrite(head,12,f2)
310 siz=head(2)/2:loop=(siz+2047)/2048
320 for i=1 to loop
330 fread(cs,2048,f)
340   for j=0 to 63
350     w=1+p(j)/63#
360     for k=0 to 31
370       cs(j*32+k)=cs(j*32+k)*w
380     next
390   next
400 fwrite(cs,2048,f2)
410 locate 0,20:print i;"/";loop
420 next
430 fcloseall()
440 endfunc
```

あとはデータを見てそのレベルより高ければ減衰させてやればよいだけ，処理はいたって簡単です。ディストーションに入る音はたいていブーストされているので，音量も上げて「飛び出した部分をばっさり切る」というのは簡単にディストーション効果を得るよい手段だとわかるでしょう。

●コンプレッサ

ダイナミックレンジ（音量レベルの差）が広すぎる音をまとめるためのエフェクタです。といっても，過入力があったときだけ音量を下げるわけにもいきませんので，音のエネルギーを時間軸方向に分散してやるような処理になっています。結果的に，アタック音が持続するのでコンプレッションサスティナーとも呼ばれます。

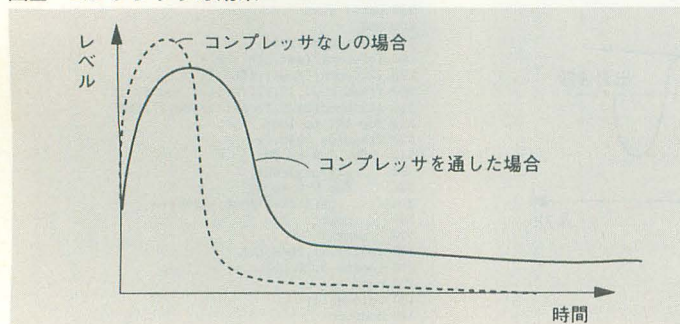
元はダイナミックレンジを抑えるためのものだったようですが，現実には副作用のほうが用途の主流になっているようです。

・実装

主体となる処理はディストーションと同じような感じですが，特性が「悪い」アンプではなくて，「そのような特性を持たせた」アンプの処理ということになります。

なにより違うのは，ディストーションが信号レベルでの処理だったのに対し，コンプレッサは音量に対する処理だということでしょう。

音量は振幅から推定していきます。これは以前作ったエンベロープ抽出プログラムと同じ処理でかまいません（1994年7月号参照)。

スレッショルドレベルを決めるのはディストーションと同じですが，さらにアタックタイム，リリースタイムというのを設定してやります。アタックタイムは鋭い信号の拾いすぎを防ぐためのもので，スレッショルドに達してから一定時間経ってはじめて動作するような機構のためのパラメータです。同様にスレッショルド以下になったときにどのくらいの時間で元の信号レベルに戻るかというのがリリースタイムになります。プログラムは省略します。

●ノイズゲート

コンプレッサと逆の働きをするものをエキスパンダといいます。音量レベルで一定の範囲だけアンプの増幅率を上げてやる感じのものです。さらにノイズゲートというものもあり「一定レンジ以下の音は削る」という処理を行います。これはさらに単純化され，ON/OFFスイッチとみなしてほかの楽器の音量で制御する使われ方をします。要するに，ある楽器の音量制御をほかの楽器のものと同じにするわけですね。

●エキサイタ（エンハンサ）

ぼやけた音をくっきりさせ，音を前に出すエフェクタです。位相を制御して音場を前に出す方式と，高域成分を強調したものをミックスして音を明るくする方式の2つがあります。処理はまったく違いますがどちらもエキサイタまたはエンハンサと呼ばれています。

・実装

位相制御はちょっと無理なので高域を強調したものを原音に混ぜてやります。これは逆FFTしたものを適当にイコライジングして作ります。あとは合成だけですね。プログラムは省略します。

●リングモジュレータ

波形同士の演算は合成作業が主体になるため，足し算が基本になります。通常，掛け算は音量やエンベロープの演算の際にしか使われることはありません。例外的に波形同士で掛け算を行うのがリングモジュレータです。

これは結果的に入力された2つの周波数の和と差をあわせた信号になります。イメージが湧かないと思いますので例を挙げましょう。一方が700HzのSIN波でもう一方が300HzのSIN波だった場合，出力される波形は400HzのSIN波と1000HzのSIN波をあわせたものになります（700−300＝400, 700＋300＝1000)。

聞いた感じ，場合によっては足し算したときとあまり変わらないこともありますが，たいていの場合は不協和音を多量に含んだ鐘のような音になります。

・実装

波形を2種類用意し，離散信号のまま掛け算して出力します。

ディレイ系のエフェクタ群

たとえば，かなりMIDIに詳しい人でもエコーとディレイとコーラスとリバーブの違いを聞かれて正確に答えられる人はそうはいないでしょう（それぞれ全部違うということなのですが，ディレイとエコーの本質的な違いはないみたいです)。そういった関係のエフェクタを大雑把にまとめてみましょう。

●エコー/ディレイ

ディレイは入力された信号を遅延して重ねるものの総称です。遅延時間が0.2〜1秒程度で単純なものがエコーということになるみたいです。ディレイはもっと多彩な処理のものをも包括した処理を指します。一度重ねた音をフィードバックして処理することもできます。エコーの場合は反響音にローパスフィルタを入れることが多いようです。

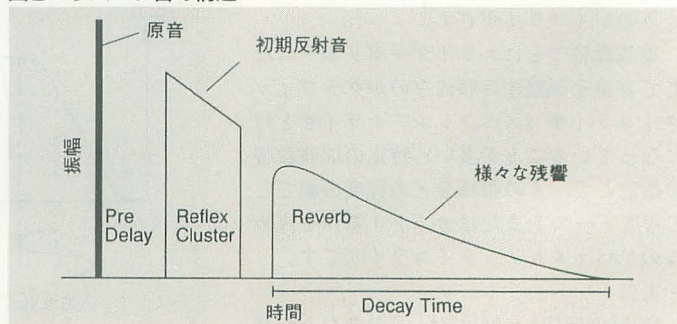
●コーラス

ディレイの特殊な状態ですが，大雑把にいえばディレイの遅延時間が不均一なものです。出力の遅延部分のテンポずれが目立たなくなり，クセが少なくなります。フィードバックはしないことが多いようです。設定するディレイタイムは13〜50msくらいで，ディレイタイム自体に不規則なLFOなどをかけてゆらぎを作ることで実現されます。ステレオの場合は左右のディレイタイムを微妙に変えて，LFOの位相を左右で反転しておきます。

●リバーブ

残響をイメージして構成されたものです。だいたい図3のような構成になります。アーリーリフレクション（初期反射）までの遅延時間で反射壁までの距離，アーリーリフレクションの大きさで壁の状態，残響の

図2 コンプレッサの効果

図3 リバーブ音の構造

量で部屋の容積，残響の音質で壁の材質を表現することができるとされています。

●フランジャー

わずかに遅延した音をフィードバックしていくことで櫛形の周波数特性を持たせたエフェクタです。ディレイタイムは0～6.5ms程度と短く，こういった範囲内でディレイタイムパラメータに三角波のLFOをかけます。ストリングス系のサウンドと相性がよいとされています。

・実装

内容はざっとこんな感じですが，ディレイを基本として，ディレイタイムの設定その他でいろいろと性質の違った効果が表れてくるのがわかります。

PCMを加工する場合，送らせた信号を小さくして重ねるというのが難しくないことはわかると思います。

また，原音Aがあったとき，

A" ＝A＋A'

A"' ＝A＋A'＋A"

A"" ＝A＋A'＋A"＋A"'

⋮

のように処理することでフィードバック効果も得られます。

あとはLFOの処理やディレイタイムの管理くらいでしょうか。

残響音は畳み込み演算で作成します。インパルス特性を示すデータ生成については1994年７月号の西川氏の記事を参照して作成してください。このときノイズの音量や長さを変えたものをいく通りか用意しておくとよいでしょう。

また，サンプリングレート15.6kHzの場合なら，ディレイタイム１msはだいたいデータ16カウント分に相当するわけですから，それだけずらしたデータを重ねていけばよいだけですね。

さて，こういったものになると，さすがにX-BASICだけでは処理できません。問題になるのは合成処理くらいですので，そういった作業は実はZPCNVなどでやったほうが遙かに効率的です。これまで不明確だったパラメータの意味などをよく考えて設定してみてください。

人魚のような音

シンセに積まれているような有名どころのエフェクトはすでに挙げてみましたが，世の中にはもっと変わったフィルタを持ったものもあります。いろいろな資料を探していて，「おや？」と思ったもののなかに「人魚のような音」という表現がありました。

それはサウンドスプライシングという技なのですが，要するにある音のアタック部分とほかの音のボディをつなぎあわせるというものです。気がつけば「ああ，そういう意味か……」というようなものですが，そういえばLA音源などもこんな感じの音づくりでしたね。これはアタック部だけ取り出した音にアタック部だけを取り除いた音をフェードインしていくことで実現できます。もちろん音程は同じでなければなりませんが。

もっと一般的な意味で，時間とともに音色を変化させるものは多くのシンセサイザで使われています。また，エフェクトのパラメータもエンベロープを持って（つまり時間で変化しながら），表現することでより多彩な音色を作り出すことができます。

その最たるものがE-MUのMorpheusから搭載されたZプレーンフィルタでしょう。

Zプレーンという３次元の空間上での音色のモーフィングを可能にしています。これは図４を見てください。

よく見ると，音のモーフィングといっても原波形をいじるのではなく，フィルタ特性をモーフィングしていることに気づきます。キーノートによる音程の変化と音量の変化は当たり前で，フィルタの波形が加わっただけ……のように見えますが，この３次元へどんな意味を割り当てるかは各フィルタごとに異なっています。３つのパラメータで幅広い操作を可能にしたなかなか賢いフィルタといえるでしょう。

最後に

今回はほとんどプログラムも提示せず考え方だけに終わってしまいました。もっともこういった操作がリアルタイムにできる環境が整っていないため，時期尚早な話題だったという説もありますが。

ちょうど先月号がCPUの特集で，CPUの違いは決定的な要因にならないと書きましたが，実際，CPU単体でできることなどたかが知れています。私などは次世代機の「CPUがなんであるか？」よりも「DSPがいくつ搭載されるか？」のほうに関心がいってしまいます（うーむ，DSP搭載を前提にしてしまっているなあ)。音源まわりで２つくらい占有できればすごく強力なんですけど……。

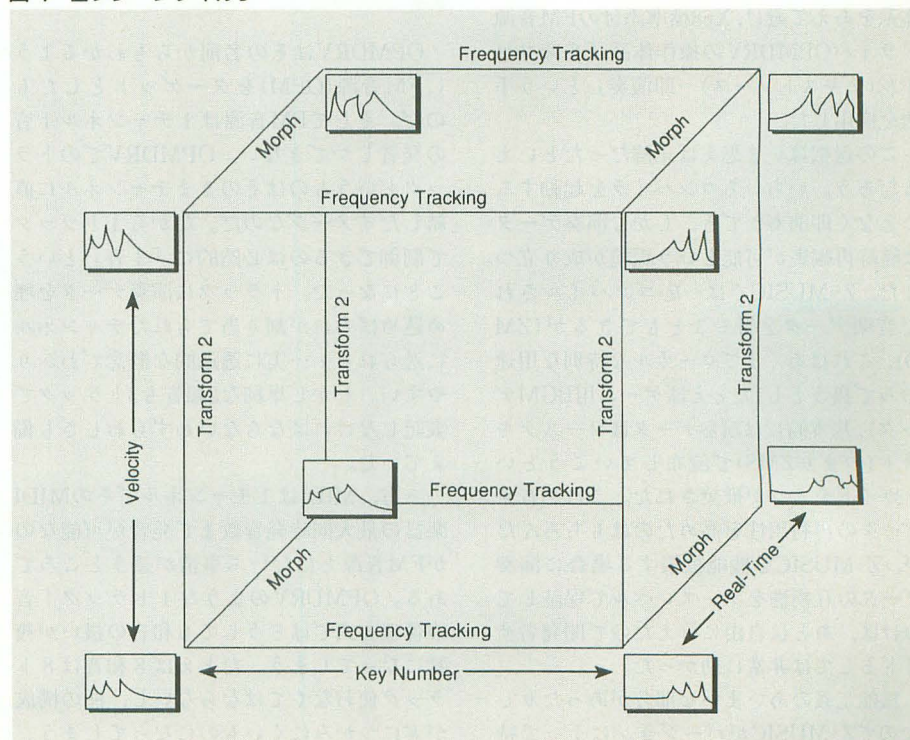
DSPがあれば今回はいい加減にすませたデジタルフィルタ部分をちゃんと作ることもできますし，さらに音場設計といった分野にまで手を伸ばすこともできるでしょう。いずれはそういったことまで考えたコンピュータミュージックを展開するようになってくるのでしょうか。

参考文献

GuitarMagazineMooks「The Effects Book」，リットーミュージック

UltraProteus Operation Manual, E-mu Systems,inc.

図4 Zプレーンフィルタ

［特集］SoundEffects

次世代システム完成間近

Z-MUSIC ver.3.0の概要

Nishikawa Zenji 西川 善司

MML式としては究極に近づき，さらにステップ式表記にも対応可能
いっそうの処理の効率化で大幅に拡張された機能も軽々とこなす
48チャンネルMIDI出力，AD PCMが旋律を奏でる……限界を超えた新システムは目前だ

初めにOPMDRV.Xありき

現在広く使われているZ-MUSIC ver.2は，基本的な機能のみを装備したZ-MUSIC ver.1をユーザーからの声をもとに強引に拡張してきた感じのシステムだった。そのためかいまになって内部を見直すと結構ツギハギの当てられた構成をしていることに気づく。

このver.2の元になっているZ-MUSIC ver.1はX680x0で扱える音源たちをなんとか1つのミュージックシステムで統括的に扱えないかというテーマの下に開発されたものだった。使いやすさという点を考慮して，フリーソフトによく見られるミュージックドライバ＋プレイヤー＋コンパイラというメジャーな構成による「コンパイラ→演奏オブジェクト生成→演奏」という操作体系をあえて避け，X68標準添付のFM音源ドライバOPMDRVの操作体系「音楽ファイル(テキストソース)→即演奏」という手法を採用した。

この選択はいま思えば正解だったといえるだろう。いちいちコンパイラを起動することなく即演奏ができ，しかも演奏データは随時再編集が可能という環境が成り立つ。また，Z-MUSICでは一応コンパイルされた音楽データを得ることもできるが(ZMD)，これはあくまでローカルな特別な用途のみで扱うとし(たとえばゲーム用BGMデータ)，基本的には演奏データはソーステキスト(つまりZMS)で流布していこうというガイドラインが推奨された。これが音楽データの再利用性を高めたのはもちろんだが，Z-MUSICを機能拡張する場合に演奏データの互換性をソースレベルで保証しておけば，あとは自由に行えたので開発者サイドとしては非常に助かった。

機能定義のあいまいな部分があったりしたのでZ-MUSICがバージョンによって特殊なケースにおける演奏が微妙に違ったり，バージョン番号チェックの曖昧さから，古いZMDを演奏すると暴走してしまうなどの失敗談はあったが，「ver.1発表→ver.2」はまずまずの成功を収めたと感じている。

で，時は流れるもの。人間の思想や世の風潮も変化していく。ver.3の設計ではいくつかの問題が挙がってきた。このページでは現在完成間近のZ-MUSIC ver.3の開発に突入するまでのプロセスをOPMDRV，Z-MUSIC ver.1/2の抱えた問題点と絡めて紹介し，後半にはver.3のウリとなる新機能を簡単に紹介したいと思う。前半は話がちょっと難しいが，今後ミュージックシーケンスソフトなどを制作される方には少し役立つ話だと思うので興味のある方はぜひ参考にしていただきたい。

1トラック1音処理の限界

OPMDRVはその名前からもわかるようにFM音源(OPM)をターゲットとしたものだ。そしてFM音源は1チャンネル1音の発音しかできない。OPMDRVでのトラックというものはそのままチャンネルに直結したイメージなのだ。だから1トラックで制御できるのは必然的に「1音」ということになった。トラックに演奏データを埋め込めばそれが割り当てられたチャンネルに送られる……実に透過的な概念でわかりやすい。しかし単純な3和音も3トラックで表記しなければならないわずらわしさも備えていた。

一方，MIDIは1チャンネルでそのMIDI楽器の最大同時発音数まで発音が可能なのがFM音源とはだいぶ事情が違うところである。OPMDRVのような1トラック1音の管理方式ではどうしても和音の扱いが複雑になってしまう。たとえば8和音は8トラック使わなくてはならないし，曲の構成が実につかみにくいものになってしまう。楽譜でなら五線譜1段で書ける4和音コードパートも，この方式だと五線譜4段分の単音フレーズというふうに表記しなければならないわけだ(図1)。

OPMDRVから派生したZ-MUSICも当然この問題に直面した。これに対してZ-MUSICは1トラック1音管理のまま，むりやり和音を発音可能にするという反則技で対処した。これは単純な同一の音長を持つ和音のみに対応したもので，各和音の構成音を列記し(最大8つ)，それに音長を与えるというものだ。単純な8和音までなら1トラックで処理できるというわけだ(図2.1)。さらに各構成音にディレイが指定できたため，あたかも各音が別々のトラックで非同期に鳴り始めたかのような分散和音の効果を得ることができた(図2.2)。

これで完璧かと思われたが，そうではなかった。発音ディレイの採用で鳴り始めは各音バラバラに発音時間をずらすことができるが，音長管理が1音分である以上，消音時間は一致してしまう。これがまず1点。次に，発音ディレイはひとつしか指定できないので，一定時間間隔ごとに発音していく処理しか実現できない。この2点を噛み砕いていうと，ちょっと凝った和音は1トラックでは表現できないということと，分散和音はやや機械的なきっちりしすぎた和音になりがち，ということになる。

そもそも，人間が和音を弾く場合を考えてみたい。同時に複数の鍵盤を叩いたとしても，それぞれの鍵盤が指によって押し込まれる時間は厳密にはバラバラである。鍵盤から指を離す場合も同様だ。つまり，発音/消音する音を単純に複数化しただけでは，「和音という音色」で単音を発音したのにすぎないということなのだ。さて，ではなぜZ-MUSIC ver.1/2ではこの1トラック1音管理をそのままで和音化したかというと，理由のひとつにワークの大きさの問題がある。もし，発音する1音1音に対し

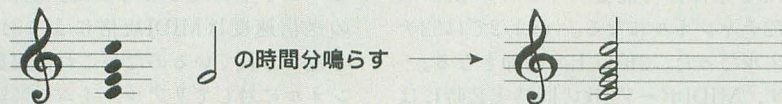
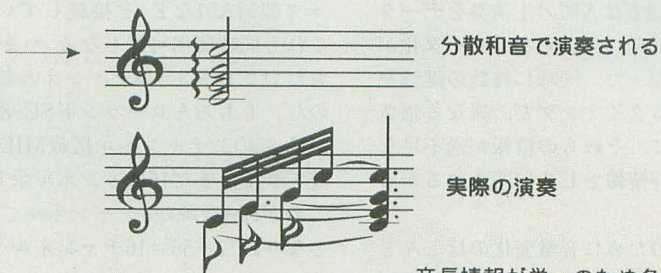
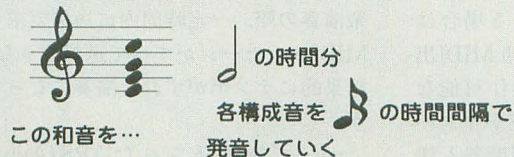
て音長管理を行うとすると，管理に非常に大量のメモリを必要としてしまう。ほとんどのトラックが結局は単音か2音程度しか鳴らされないと思われる状況で，使われもしないワークをメモリ上にリザーブしておくのはあまりにももったいない。

もうひとつの理由に処理速度があった。もし，各音に対して音長管理を行うとすると，結局管理する音長の数だけワークを処理せねばならないわけで，8和音の和音を処理するとしたら，8トラック分処理するのと大差ないことになるのだ。

この2点からZ-MUSIC ver.1は現在の手法を採択した。実際MIDIにおいても演奏データを人間がコンピュータ上で作成する場合に限ってはなんの問題もなかったのだが……。

人間の生演奏のデータ化

問題は人間が弾いたものをデータ化する場合だ。現在フリーソフトでMIDIキーボードで弾いた演奏をデータ化するソフトがあるが，Z-MUSIC ver.1/2用のデータにした場合，非常に醜いデータになってしまう。たとえばドとミの和音がドが絶対音長で48カウント鳴り，ミが47カウント鳴ったとすると，ミのほうがたった1カウント少ないだけでこの和音のデータは2トラックで表さなくてはならないのだ。両方とも48だったならばZ-MUSIC ver.1で採用したその単音管理方式の和音でめでたく1トラック表記できるのに，なんともはがゆい結果となる。

図2　1トラック1音管理方式で和音を鳴らす

ベロシティ/音量変化

OPMDRVにおける音量変化は，あるトラックのデフォルト音量を設定するという目的が大前提である。1音1音に対して音量を設定することは希であり，以前に設定した音量がどうしてもその局面にそぐわないときにはまた音量を設定する……そういう使い方を想定している。

OPMDRVの流れを汲んでいるZ-MUSICも当然このコマンド体系に倣ったが和音のときと同様に，生演奏のデータ化の際に問題が生じた。人間の指は鍵盤を同じ強さで叩いたつもりでも，厳密には微妙にその音量は違う。たとえばある演奏のある音が100の音量値で発音して，続く音が101で発音した場合，これをデータ化すると，

音量コマンド：100,発音コマンド,音量コマンド：101,発音コマンド

のようになる。もし後ろの音が1少ない100の音量値だったならば，

音量コマンド：100,発音コマンド,発音コマンド

となり音量コマンドがひとつ少なくて済むだろう。しかし，実際の人間の演奏はこのような微妙に音量のばらついた音が連続的に出現するので，これをデータ化すると音量コマンドだらけになってしまうのだ。

ver.2からver.3へ

ここまでで紹介した音長と音量の問題に対しての現時点で最善と思われる答えをver.3に反映した。これがver.3がOPMDRV, ver.1/2から大きく変わった点である。いい換えるとver.3はバージョンアップに際してOPMDRVの呪縛から解き放たれたということである(前バージョンとの演奏データのソースレベルでの互換性を保つ関係でときおり昔の制約がふと顔を出すこともあるが)。

まず，1トラックで最大16音のまったく独立した発音/消音を可能にした。詳しくいうとそのトラックで発音した1つひとつの

図1　1トラック1音方式

音符が個別に音長管理がなされるということである。前述したように，16音分を1トラックで使用するような局面は希なので「使われないかもしれないワーク」がver.2以前のと比べて増加してしまっているかもしれないが，今後の拡張性，汎用性を考えればこうするのがもっとも妥当な進化の方向と考えた。ver.1発表当時は1Mバイトユーザーを考慮する必要がどうしてもあったため，無駄なワークを少しでも抑ええようと仕様を決定していった。しかし4Mバイト実装が標準となろうとしている現在では，多少のメモリを消費しても機能拡張性が高い方針を選ぶべきと判断した。

ベロシティ(音量変化)の扱いもver.3になって内部的に改変した。「発音する1音1音が音量情報を個別に持つ」という方式になった。これには，MIDIがコンピュータミュージックシーンに浸透していく流れのなかでそのデータを扱うのにいちばん適したかたちに順応していたという意味あいがある。それではここでMIDIの世界においての音量情報の扱いについて簡単に解説しておこう。

MIDIの楽器に対する発音メッセージ(コマンド)のパラメータは，ノート番号(音高に相当)，音量(ベロシティ，ここでは以下音量とベロシティは同義と仮定する)の2つなのである。つまり，1回の発音コマンドごとに音量を指定しているといったイメージなのだ。MIDI規格というのは楽器を共通規格の下で制御するという目的で制定されているが，その制御コマンド構造は，「人間の生演奏を効率よくデータ化する」という思想の下に設計されている節がある。

ver.2設計時には1音ごとに音量情報をつけるのは演奏データにあまりにも無駄が多くなりすぎると判断したために分離して扱うことにした。これに対してver.3ではMIDIの思想を導入し，音符情報を音量情報とペアにして扱うということにしたわけだ(従来方式の設定も可能)。

音長管理，音量情報の扱い，これらの改変により，ver.3では人間の生演奏をデータ化した場合，非常に無駄のないデータ化が行えるようになった。同時に複数の鍵盤をそれぞれ異なるタイミングで，異なる強さで叩いたときに，それらの情報が過不足なくひとつの音符情報として展開されるからである。

もちろんこのために音量変化のほとんどない状況でも，音符には絶えず音量情報が付随することになるのでver.2のときよりも明らかに演奏オブジェクトデータ(ZMD)は肥大化する。しかし，人間の生演奏との融合を図るならば，この手法のほうが優れていることは説明するまでもないだろう。

ところでこういった変革がなされてはいるが，ユーザー側はそれほどその違いを意識しなくてよい。ZMSの互換が保たれている点もあってむしろ気づかないかもしれない。それはそれでまたいいことなのだ。昔ながらのユーザーが離れるよりは。実際ver.3専用の機能を使っていかないと，この変革の恩恵あるいは影響は見掛け上まったくない。保守派の人も安心なのだ。

ところで最後にいっておきたいのはZ-MUSIC ver.1/2方式がまるきし駄目だったというわけではないということ。ver.1/2と同様の処理方式を採用した音楽制御プログラムを批判しているわけでもない。いまでも1トラック1音処理系のミュージックソフトは世の中にたくさんあり多くのユーザーを獲得している。また演奏データのコンパクト性というものを考えると明らかに音量情報は別扱いにしたほうがよい。特にメモリの節約を第一に考えるゲーム音楽の方面では。あくまでver.3は進化の方向を転換したというだけなのだ。そのあたりはご理解いただきたい。

ver.3の新機能

ver.3になってさらに新機能が追加された。また従来機能もかなり拡張されている。ver.3の実際のリリースは少し先になるが，現在装備している機能はもう公開しても構わないだろう。ということで，ここからは肩の力を抜いて読んでもらいたい。

●同時制御可能チャンネル数

ver.3では操作可能なチャンネル数は最大で72チャンネルになる。ver.1/2では32チャンネルだった。2倍以上の増加となる。

まず，MIDIボード2枚の同時実装時にはそれぞれを同時に制御可能になった。さらにRS-232CにMIDI変換アダプタ(PC-98用などがそのまま流用可能/カモンミュージック製MA01など)を接続している場合はこれも同時使用可能となる。つまりMIDI出力だけで16×3＝48チャンネル操作可能なのだ。もちろんローランドSC-88やヤマハMU-80の32チャンネル搭載MIDI楽器も接続できて，まだ16チャンネル余る。・

あとFM音源が8チャンネルで48＋8＝56となり，72－56＝16チャンネル……あと16チャンネルはいったいなにか。

●新開発AD PCMドライバ搭載

ver.3は残念ながらPCM8.Xには対応していない。その代わりといってはなんだが，専用のAD PCMドライバがver.3スタッフの手によって開発された。X680x0の内蔵AD PCM音源に対して同時発音数16(効果音再生専用チャンネルを別に持っているので正確には17)，リアルタイム音程変換，128段階音量可変機能を装備している。

音程変換可能ドライバは実は世の中にすでにあるにはあったのだが演奏する際に16ビットPCMデータなどに変換するなどの前処理が必要でわずらわしい面があった。しかし新しいドライバでは通常のAD PCMデータがそのまま使用可能で，特別な前処理はなにもいらない。普段鳴らしているAD PCMデータをそのまま扱えるのだ(CPUパワーの消費量は凄い……)。このドライバを駆使してAD PCMチャンネルでポルタメント，ピッチモジュレーション，音量モジュレーションなどが実現できるようになった。ver.3ではAD PCMがOPMのようにシーケンスできるようになったというわけだ。

また現在，瀧氏がかなり高機能なX68000用周辺ボードを制作しているが，このver.3用AD PCMドライバはこのボードのPCM機能スペックを想定したインタフェイスを内蔵している。よって将来，PCMドライバをこのボード対応のものに差し替えるだけで従来の演奏データがこのボードを用いて高音質で演奏できる……ということになっている。瀧氏よ，急げ……。

●新MIDI送信システムVTMS/ARS搭載

大げさで申しわけない。

MIDI楽器を制御する際に，その制御にMIDIメッセージを送信しているのだが，いままでのミュージックソフトではそのメッセージを垂れ流ししていた。現在のMIDIの送信速度はMIDI規格により31,250bpsと定められているのだがこれは複数のチャンネルに対してリアルタイムに楽器を制御するにはやや不十分な速度なのだ(MIDI規格は10年以上も前の古い規格で当時の実用技術範囲内のものだからしかたないが)。音楽演奏の際，一定時間内に送信要求されたMIDIメッセージがすべて送信できないと，結果的にテンポがずれた演奏になってしまう。

ver.3で新開発されたARS(Automatic Running Status)とは楽器に送るMIDIメッセージをリアルタイムに最適化して送信するシステムだ。これでいくらか送信量を減らすことができる。

もうひとつのVTMS(Variable Transmit Mode System)とはMIDIインタフェ

イスの負荷のかかり具合によりその送信手法を切り換えるというもの。CPUの負荷が軽いときも重いときもMIDIメッセージの送信が平均的に実行される効果が得られる。これを採用したver.3ではver.1/2よりもさらにテンポずれが起こりにくくなっている。

●ARCCの機能拡張

ver.2ではひとつのトラックに対してひとつのARCCを実行できるだけだったが，ver.3では4つまでが独立して機能する。

ARCCとはMIDIにおいて任意のコントロールをユーザー波形や算術波形を用いてモジュレートできるというものであった。たとえばARCC機能で音量に対して周期的増減変化を与えればトレモロ効果が実現できた。工夫次第でMIDI楽器に装備されていない演奏表現も可能にするZ-MUSICの特徴的な機能だった。

しかし1トラックで使用できるARCCの数はひとつであったため，トレモロ効果とワウワウ効果を実行したり，複数の効果をひとつの音にかけることはできなかったのだが，ver. 3では最大4つの効果を同時に与えることができるようになったわけだ。

さらに，FM音源のアンプリチュードモジュレーションもver.3からはARCCという名称に統合された。FM音源トラックに対しても4つのARCCを機能させることができるようになったのだ。ARCCとして制御できるのは各オペレータのトータルレベル，ハードLFO，パンポット，ノイズなどなど。FM音源の4つのオペレータ1つひとつに4つのARCCを機能させ，それぞれ違った波形でモジュレートすることも可能だ。どんな音が出てくるか想像もつかない。

今回搭載されたFM音源対応のARCCは，FM音源の新しい使い方を生み出すかもしれない。

●アゴーギク

演奏速度に緩急をつけることを音楽用語でアゴーギク(Agogik)という。ver.3では大胆にも，テンポをユーザー波形や算術波形をソースに緩急をつける機能アゴーギクが搭載された。平たくいうとテンポに対するARCCとかモジュレーションのようなものだ。使用価値は未知数だが。

●エンハンストベロシティシーケンス

ベロシティの変化をこれまたユーザー波形や算術波形でつけようというものだ。これも平たくいってしまえば，ベロシティに対するARCCとかモジュレーションのようなものだ。

●コンパイラの分離

ver.2では正規版のZMUSIC.Xとコンパイラなし版であるZMSC.Xの2つが存在したがバージョンアップの管理が困難だった。そこでver.3ではコンパイラと演奏制御部分をそれぞれの独立したプログラムに解体した。ver.3のZ-MUSIC本体はver.2でいうコンパイラなし版に相当することになる。

分離してしまったとはいえ，両方常駐することによりもちろん従来通り演奏データはソース(ZMS)レベルで即演奏が可能だ。

また，Z-MUSIC ver.3で規定されたインタフェイスを持ったプリプロセッサやほかの言語記述方式のコンパイラを開発すればこれが組み込める。つまりZMSの文法以外のファイルなども(たとえばNAGDRVやMXDRVのソースなど)直接演奏することもできてしまう(かもしれない)。

●バッファ概念の撤去

ver.3はAD PCMバッファやトラックバッファなどを一切確保する必要がない。必要なとき必要なだけメモリを確保し提供する。このためいちいち各バッファの容量を変更して再常駐するような状況から解放される。

また，特筆すべきなのはm_alloc()命令の撤廃だろう。ver.2以前では演奏データを格納するトラックの容量をあらかじめユーザーが想定して，その容量を設定しなければならなかったが，バッファ概念がver.3ではまるごとないのでそういった手続きも不要になった。

●制限の撤去

音楽反復記号のCodaやD.S.などの多重定義が制限なしになった。またver.2では最大8重まで組むことができた多重ループも何重まででもOKとなった。

モジュレーション/ARCC/アフタータッチシーケンスの音長1/8モードが拡張され，音長の1/8以外の間隔の振幅の変化が可能になった。このため演奏に，より自由度の高い情緒変化を与えられるようになった。

●その他

モジュレーション関係の波形にも機能拡張がなされた。まずプリセット算術波形にノイズが加わった。それにユーザー波形(波形メモリ)に対して振幅を与えられるようになった。モジュレーション実行中の任意のタイミングで波形種類の切り換えが可能だが，このとき，接続時に音が不自然にならないようにスムージングが自動的に行われようになっている。

同一音を複数回押すような挙動をシーケンスすると自動的にこれをアフタータッチとして認識，ポリフォニックプレッシャーメッセージを送信する。ポリフォニックプレッシャーに対応したミュージックソフトはまだあまりないのが現状。これも使用価値が未知数の機能だ。

音楽テンポの始動源としてver.2以前はOPMタイマを使用していたがMIDIボードを接続している場合はこれに搭載されている14ビット高精度タイマを使用するようにした。この場合はOPMタイマを使用したときよりも誤差の少ないテンポが継続できる。また，音楽演奏と効果音はまったく無関係のテンポを設定可能になった。ver.2以前の効果音モードとは違い，通常の演奏データも効果音として即演奏が可能なので，やろうと思えば同時にまったく別の2曲を演奏することもできる(あまり意味のないことかもしれない)。

*　　*　　*

いまのところver.3の最大の魅力はなんといってもMIDI48チャンネル同時制御にあると思う。いまからRS-232C MIDIモジュールや，もう1枚MIDIボードを購入しておいてくれ。SC-88に買い換えたり，もう1台MIDI音源を買ったりするのもいいかもしれない。

では乞うご期待。

ver.3用AD PCMドライバとは

Z-MUSIC ver. 3用に作成されたAD PCMドライバだが，名称はMPCM.X。同時発声数16，リアルタイム音程音量変換が可能。よって，AD PCMパートでもポルタメントやピッチモジュレーション/アンプリチュードモジュレーションなどが実現可能である。

また，ループ領域を設定するとその区間を自動的にループ再生する機能を備えている。これによって，たとえばストリングスなどを発音する場合，持続音をループ指定しておけば任意の時間再生可能となるわけだ。これまでは長い音を鳴らそうとすると巨大なAD PCMデータを必要としたが，これが改善されるだろう。

と，かなり売り文句を連ねたがこのMPCMはかなりメモリとMPUパワーを消費する(常駐サイズ約200Kバイト)。メモリは増設すればよいが問題はMPUパワーだ。現実的な話，10MHzマシンでは音程をリアルタイム変化させた場合，実用的な動作ができるチャンネル数は1,2程度だ。16MHz機だと4,5くらいか。X68030だと8前後。まあ，音程音量変換しないチャンネルがほとんどであろうから，もうちょっと発音はできる……にしても10MHzにはきついか（アクセラレータに期待か？）。音程音量変換しない場合はだいたいPCM8.Xより少し速い程度のパフォーマンスを見せてくれている。

新システムはX68000の内蔵音源部を究極的にパワーアップする予定だ。

Oh!X LIVE in '95

X68000・Z-MUSIC
ver.2.0(SC-55対応)

魔法のプリンセスミンキーモモより
ラブラブミンキーモモ

Sakamoto Makoto 坂本 誠

X68000・Z-MUSIC
ver.2.0+PCM8用

©1988スクウェア ファイナルファンタジーIIより
メインテーマ

Akeno Hiroyuki 明野 浩之

X68000・Z-MUSIC
ver.2.0(SC-55対応)

ショパン練習曲第3番ホ短調 Op.10-3
別れの曲

Takahashi Toshiyuki 高橋 利之

X68000・Z-MUSIC
ver.2.0(SC-55対応)

宇宙戦艦ヤマト完結編より
ルガール総統の戦争

Hayasaka Makoto 早坂 真

音楽特集ということで，今月は豪華版のLIVE in。アニメの曲からゲームミュージック，クラシックといろいろ取り揃えてみました。どれもなかなかの技師たちの力作です。ひとつ気合を入れて聴いてみてください。

大人になったらなんになる

最初はテレビアニメ「魔法のプリンセスミンキーモモ」のオープニングテーマです。

曲は内蔵音源＋MIDI(SC-55系)という構成をとっていますので，ミキサーやSC-55の外部入力端子を使用してミキシングしてください。

このようなアニメソングの場合，ボーカルを歌っていた歌手の声の印象が聞き手に強く残っている場合が多いので，ボーカル(メロディ)パートの音色選択には非常に苦労するものです。どうしてもあわない場合はメロディなしのカラオケバージョンにしてしまうのが安易でいいでしょう。さて，今回の曲データではボーカルパートをポップなオルガンで綴っていますが，これがなかなかいい味を出しています。原曲のボーカリストの声の雰囲気も出ていますし，なんといっても他パートとのハーモニーが非常に美しく聞こえます。

それとこの曲データに関して特筆すべきはアコースティックギターです。なんとFM音源で鳴らしているのですが，非常にリアルに聞こえます。単音鳴らしただけではそうでもないのですが，曲中では実に本物っぽい音がしています。きっとシーケンスの手法(曲データの打ち込み方)がよいからなのでしょう。興味のある人はリストを見てみましょう。

ミンキーモモ

ファイナルファンタジーII in SFC!?

次の曲は内蔵FM音源とAD PCM音源をフルに稼働させた意欲作「ファイナルファンタジーII」グレードアップアレンジバージョン。一世を風靡したファミコンRPGのパート2のメインテーマです。原曲は当然ゲームが初代ファミコン上のものだったのでPSG数声のシンプルなものです。これをデータ作者の明野君は，このゲームがスーパーファミコン上のものだったらきっと曲はこんなものになっただろうという想定でイメージを膨らませてアレンジを進めたそうです。なるほど，バックのディレイエフェクトを効かせたスネアパート，細かく刻まれたフレットレスベースパターン，ストリングスの被り方はスーパーファミコン版で登場した「ファイナルファンタジーV」の曲のアレンジに非常に通ずるものがあります。メロディがダブルリードとかフルート系の音色になっているあたりも同シリーズのゲームのファンの方なら思わずニヤリなのではないでしょうか。

さて演奏にはZMUSIC.X ver.2.0とPCM8.Xのほかにフ-MUSICシステムver.2.0に同梱のAD PCMデータとZPCNV.RとZPLK.Rが必要です。まずリスト5のバッチファイル「FF2MTAR.BAT」と「FF2MTAR.CNF」を入力してください。次に必要な全AD PCMデータとZPCNV,ZPLKにパスが通っている環境で，

A>FF2MTAR.BAT　(リターン)

ファイナルファンタジーII

としてバッチファイルを実行してください。ZPDは500Kバイトを超えるほど巨大なものになるのでメモリとディスクの容量が相当量必要です。ZPDができているのを確認したら曲データ「FF2MTAR.ZMS」を入力し，PCM8.XとZMUSICを常駐させて，

A>ZP FF2MTAR.ZMS　(リターン)

で演奏できます。

ピアノコンサートへようこそ

1994年12月号で生演奏さながらのショパン「幻想即興曲」を聞かせてくれた高橋氏が今度は「別れのエチュード」を投稿してきてくれました。今回もペダルワーク，各キーの音圧の変化が非常にリアルです。テンポも局面ごとに変化していき，演奏者の情緒の変化さえも演奏データ化している印象を受けます。

プログラムリストは音符情報の可読性を考慮した関係か，ペダルワークのトラックが演奏トラックから分離してありますね。MIDIを扱うときにこういう手法はなかなか頭のいいやり方です。複数のトラックを1つのチャンネルに割り当てて，あるトラックではエフェクト，あるトラックでは実際の演奏……このようにすれば演奏トラックが見やすくなるだけでなく，音符の音長にとらわれない自由なエフェクトが設定できるのです。

演奏はすべてのGS音源で行えます。いうまでもなく使用音色がピアノだけなのでお手持ちのMIDI楽器やMIDIピアノにも簡単に移植ができると思います。

ヤマトの彼がヤマトで再登場

1994年4月号で「宇宙戦艦ヤマト一誕生一」でDTMとは思えない演奏をぶちかます驚異のデータを発表，そして1994年12月号で「きまぐれオレンジ☆ロード」のオープニングテーマでもやはり期待に応えてくれた，あの早坂君の登場です。今回は再びヤマトネタを送ってくれました。「宇宙戦艦ヤマト完結編」から「ルガール総統の戦争」です。かなり選曲が渋めですね。

タイトルから想像できるように今回は緊張感あふれる戦闘時のBGMです。実際，管楽器，弦楽器が演奏中，テンポの速いフレーズをぶつけあって戦います。これを背後で盛り上げるのがティンパニのリズム。交響楽の醍醐味ですよね。

以前このコーナーでお話しした異楽器間のユニゾンから生まれる「音の色」ですが，この演奏でも効果的に使われています。序盤，迫力ある太い音がたくさん登場しますがこれはティンパニを含めさまざまな楽器が一斉に鳴って作り出された音色です。また途中でもさまざまな楽器がユニゾンフレーズを展開しますので，MIDI楽器のパネルを操作したり，モニタリングツール(ZAMなど)を使用してどの音色とどの音色がユニゾンしているのかを確かめながら聴くと曲作りの勉強になると思います。

別れの曲

演奏にはGS系音源が必要です。SC-55/SC-55mkIIいずれにおいても正常な演奏を確認しました。

また今回早坂君はおまけとして宇宙戦艦ヤマトの艦内警告音のリストをつけてくれました。演奏してみてください。笑えます。

ところで以前早坂君が本コーナーで発表した「宇宙戦艦ヤマト一誕生一」のオリジナル演奏が収録されている「交響組曲・宇宙戦艦ヤマト」ですが，CDとして最近再版されました。なかなかの名曲/名演奏ぞろいでおすすめです(日本コロムビアCD:COCC-1227　2,700円(税込))。

リスト1　ラブラブミンキーモモ

日本音楽著作権協会(出)許諾第9472828-401号

```
 1:.comment ラブ・ラブ・ミンキーモモ (TV-SizeVer.) By まーひー
 2:/(C) 小山茉美・荒木とよひさ・佐々木勉・桜庭伸幸
 3:/Programmed '93/11/20～24,94/12/15
 4:
 5:/Z-muSiC init.
 6:(i)
 7:(d0)
 8:
 9:/ Track setup.
10:(m1,2000)(aMidi1,1) / Melody
11:(m2,2500)(aMidi2,2) / Bass
12:(m3,4000)(aMidi3,3) / Rhodes
13:(m4,2000)(aMidi4,4) / Strings(Left)
14:(m5,2000)(aMidi5,5) / Strings(Right)
15:(m6,2000)(aMidi6,6) / E.G.(L&C)
16:(m7,2000)(aMidi7,7) / E.G.(R)
17:(m8,2000)(aMidi8,8) / Syn.Hit
18:(m9,2000)(aMidi9,9) / Chorus
19:(m10,2000)(aMidi10,10)       / Drums
20:(m11,2000)(aMidi10,11)
21:(m12,2000)(aMidi10,12)
22:
23:(m13,2000)(aFM1,13)
24:(m14,2000)(aFM2,14)
25:(m15,2000)(aFM3,15)
26:(m16,5000)(aFM4,16)
27:
28:/-----------------------------------------------------
29:/ VOICE SET
30:/*       AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  MiyoSyn.
31:(@71,    26, 12,  0,  7,  5, 21,  1,  0,  3,  0,  0
32:         24, 10,  0,  9,  5,  0,  0,  1,  3,  0,  0
33:         26, 12,  0,  7,  5, 11,  1,  0,  4,  0,  0
34:         24, 10,  0,  9,  5,  3,  0,  1,  4,  0,  0
35:/*       AL  FB  OM PAN  WF SYN SPD PMD AMD PMS AMS
36:          4,  6, 15,  3,  0,  0,  0,  0,  0,  0,  0)
37:
38:/*       AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Square
39:(@72,    31,  0,  0,  0,  0, 26,  2,  2,  0,  0,  0
40:         26,  0, 12,  6,  0,  0,  0,  1,  0,  0,  0
41:         26,  0, 12,  6,  0,  0,  0,  1,  0,  0,  0
42:         26,  0, 12,  6,  0,  0,  0,  1,  0,  0,  0
43:/*       AL  FB  OM PAN  WF SYN SPD PMD AMD PMS AMS
44:          5,  7, 15,  3,  0,  0,  0,  0,  0,  0,  0)
45:
46:/*       AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  FolkGt
47:(@73,    21,  7,  8,  2,  3, 36,  1,  1,  0,  0,  0
48:         27, 18,  7,  6,  4, 23,  2, 12,  0,  0,  0
49:         25,  6,  4,  2,  2, 36,  2,  4,  0,  0,  0
50:         31,  8,  6, 10,  1,  0,  2,  1,  0,  0,  0
51:/*       AL  FB  OM PAN  WF SYN SPD PMD AMD PMS AMS
52:          2,  7, 15,  3,  0,  0,  0,  0,  0,  0,  0)
53:
54:/-----------------------------------------------------
55:/ INIT
56:.sc55_init $10
57:
58:.sc55_v_reserve={2,1,4,2,2,4,1,4, 3,1,0,0,0,0,0,0} /20+4=24
59:
60:/                    Mac Chr PreLLev TimeDlFBToCho
61:.SC55_reverb $10={  2,  5,  5,127, 84, 50,  0}     / Rev. set
62:/                    Mac PreLLev FB  Dly RateDeptToRev
63:.SC55_chorus $10={  0,  0, 40,  2, 20, 72,  1,0}   / Cho. set
64:
65:.sc55_print "Love Love MINKY MOMO By M.Koyama"
66:
67:(t1,2,3,4,5)
68:     @g12 @I$41,$10,$42
69:(t6,7,8,9,10)
70:     @g12 @I$41,$10,$42
71:
72:(t1,3,6,7,9)
73:     [K.Sign -b]
74:(t13,14,15,16)
75:     [K.Sign -b]
76:
77:(t1)  @E40,48 @X$b0,1,32     / GS ID & Eff.Bal.
78:(t2)  @E40,00
79:(t3)  @E40,48
80:(t4)  @E127,32
81:(t5)  @E127,32
82:(t6)  @E96,96
```

```
 83:(t7)  @E96,96
 84:(t8)  @E84,00                    x$40,$18,$14,0  /Single assign
 85:(t9)  @E40,48 @Y1,$66,54x$40,$18,$14,0          /Single assign
 86:(t10) @E48,00                    x$40,$10,$14,1
 87:     @Y$1d,41,0         /Toms    Rev send(off)
 88:     @Y$1d,43,0
 89:     @Y$1d,45,0
 90:     @Y$1d,47,0
 91:     @Y$1d,69,0         /Cabasa
 92:
 93:(t1) T155
 94:
 95:(t1) @18v16           R4
 96:(t2) @34v15@y1,32,44 @y1,$63,54 @y1,$64,74 / CF,AT,DK
 97:                     R4
 98:(t3) @05v10@p80                 R4
 99:(t4) @49v12@p32
100:      @y1,$63,59                R4        / AT
101:(t5) @49v12@p96
102:      @y1,$63,59                R4
103:(t6) @30v14"1                   R4
104:(t7) @30v14"1                   R4
105:(t8) @101v11@p84    R4
106:(t9) @92v11         R4
107:(t10) v13@u127        .         R4
108:       @y$18,38,67              / SD:Pitch
109:(t11,12,13,14)                  R4
110:(t15) @B96          R4
111:(t16) @73p1v14                  R4
112:/----------------------------------------------------------
113:/ MML Data SET
114:
115:/ Melody
116:(t1)          r1 r1 r1 r1
117:
118:/            **I**
119:    o518|:c4g+32a..g4f4 ed4.r2 | fffff4ag 4.r2. :|
120:     dddde4f ^1 rrccd4e4
121:
122:/            **A**
123:    g4.f2r r1 ddddde4d 4rdde4f4
124:     argg2r r1 e4eeef4g 4.c4d4e4
125:    f+32g16.f4f2r r1 ddddde4d 4rc+32d..e4f4
126:     aagg2r r1 eeeeef4f+32g16. ^2.r
127:
128:/            **B**
129:(t1)         a32b ..bbbbrde 4.r2. a4c4c4ed ^2.ra32b16.&
130:     bbbbbrde 2r{edc}2 g4.a^2 ^2.rr
131:
132:/            **C**
133:(t1)         a32b16.^4d^2 rre4f4g4 ra4aga4g f+2r2
134:     b4b4b4a4 ag4.r2 e-32e16.eeefg4f ^1
135:
136:    r1 r1 r1
137:
138:/ Bass
139:(t2)         o218ffa<crc>aq5fq8 ddfaraf(a16g),6r16
140:     >ggb-<drd>b-<c rq4cq8c2.
141:
142:/            **I**
143:    o218 |:ffa<crc>aq5fq8 ddfaraf(a16g),6r16
144:             |>ggb-<drd>b-<q5dq8 ccegrgeq5cq8 :|
145:              >g4rg<ccrf r@u99|:7 fu+4 :| fr2..
146:
147:/            **A**
148:    |:f4rarq5aq8|<c4:| (c4>a),24 |:d4rfrq5fq8|a4:| (a4g),24
149:    |:>g4rb-rq5b-q8<|d4:| dr |:c4rere|g4:| (g4g-),24
150:
151:    |:f4rarq5aq8|<c4:| (c4>a),24 |:d4rfrq5fq8|a4:| ar
152:    |:>g4rb-rq5b-q8<|d4:| dr |:c4rere|g4:| (g4g-),24
153:
154:/            **B**
155:    >|:ggb-<drd>b-<d ccegrgec
156:     |>aa<cerecq5eq8 ddf+araf+(dc),12 :|
157:      ffa<c.r16c>a<c >ffa<c4e-c>(af),6
158:
159:/            **C**
160:    l4>b-r8<d.f cr8e.g fr8@q23f8q8>ar8q5a8q8 <dr8f+.a8r8
161:    l8>g4rb-4.<dr |:c4re4.gr:| f4.arq5aq8<(c4>g)
162:
163:    q6gggrccrq8f^ 1 ^1
164:
165:/ Fender Rhodes
166:(t3)         o4q318'fa<c'rrq8'f2a<c'r q3'ga<c'rrq8'g2a<c'r
167:    q3'gb<d'rrq8'g4.g<d'r 'gb<ce'rq3'gb<ce'q8'g2.b<ce'
168:
169:/            **I**
170:    q3'fa<c'rrq8'f2a<c'r q3'fa<cd'rrq8'f4.a<ce'&q3'fa<cd'r
171:     'f2b<d'q8'f4b<d'r 'g1b<ce'r
172:    q3'fa<c'rrq8'f4.a<c'q3'fa<c'r
173:      'a<ce'rrq8'a4.<cdf'q4'a<cdf'r
174:    'g2b<cf'q8|:'gb<ce':|r'a1<cf' r 'a16<cf'r2...
175:
176:/            **A**
177:    |:q6'fb<d''fa<c'r'f2a<c'r:|q8'f1^^a<d' r'f4.a<ce'q3<dr
178:    |:'ad>b'rrq8'g2d>b'r:| |:'e4.c>bg'|rq6|:'ec>bg':|r:|q8
179:     'c4>bg''d4>bg''e4c>bg'>
180:
181:    |:q6'fb<d''fa<c'r'f2a<c'r:|
182:       q8'f1^^a<cd' r'f4.a<ce'q3<dr
183:    |:q6|:'ad>b':|rq8'g2d>b'r:|
184:       'e4c>bg'r4q6'ec>bg''fd>bg'rq8'g1ec>b'r
185:
186:/            **B**
187:    >'d1gb' 'c1eg' 'c1eg' 'c2.df+a'b!+4
188:    'd2.fgb'&'d4fgb<d' 'e2g<c''c2eg'
189:     q6|:'cfg':|q8r'c4.fa'b!+4 'f1a<ce-',3
190:
191:/            **C**
192:    u-16q3|:|:8'fb<d',0:| |:8'eg<c':|
193:              ||:4'cfa':||:3'c+eg':|r |:8'df+a':|:|
194:              |:8'eg<c':||:5'fa<c':|rrr u
195:
196:    q6|:3'fb<d':|r|:'gb<ce':|r'a8^1^1<cf'
197:
198:/ Strings Left&Center
199:(t4)          r1 r1 r1 r1
200:    |:7 r1 :| r4o3l16cdfafga<c>a<cdf fr2...
201:/             **A**
202:    o618@46
203:    |:|:r2rcrr:| r1 r1 |:r2rdrr:| r1
204:       |rc>b-<c>a<c>g<c:|
205:        r4@49@p64l16o5cdefefgagab-<c
206:/             **B**
207:    l4d1 ag2. c2eg gf+2^8r8  >b-2.<d crq4>{efg}2q8
208:    g.a.<c e-2^8r8{e-dc>b-agfe-}4
209:/             **C**
210:    u-16@p32d2.f agre c2c+2 d>ab-q4aq8
211:    g2f2 er.q4e8f8f+8q8 g2b-2 a2.r
212:
213:    r1 r1 r1
214:
215:/ Strings Right
216:(t5)          r1 r1 r1 r1
217:/             **I**
218:    |:7 r1 :| r4o5l16cdfafga<c>a<cdf fr2...
219:/             **A**
220:    o518@46
221:    |:|:r2rcrr:| r1 r1 |:r2rdrr:| r1
222:     |rc>b-<c>a<c>g<c:|
223:      r1
224:/             **B**
225:    |:8  r1 :|
226:/             **C**
227:    u-16 @50o4l1'd>b-' 'e>b-' a2g2 f+4&'f+2cd'r4
228:    'd>g' 'c4>g'r4.@49q4{cdd+}4. @50q8'e2>b-''d2g' 'f2.c'r4
229:
230:    r1 r1 r1
231:
232:/ E.G.(L&C)
233:(t6)          @p48o518 @H32@M96
234:(t6)          (ga)&a2^16&@B0,-1366a16@B0gf g(ga)&a4.&@M@B0,-2732
a@B0r
235:(t6)          (c+16d) &d8.d{cd}8fgfr<@M96c 4cc2^&@M(c>e)
236:/             **I**
237:    |:8 r1 :| r1
238:/             **A**
239:    |:16 r1 :|
240:/             **B**
241:    @29@p64u-32o3l16
242:    |:gr{gb<d}8grdrugrdru-32>brgr rr'g<c',1rr4'g8<ceg'r4.
243:    |  ar{a<ce}8areruareu-32rcr>ar r8<'cf+'rr4'c8f+a'r4.:|
244:    crcfarfr<ucu-32r>arfrcr rr'e-g'rr4'e-8g'r4. u@30
245:/             **C**
246:    |:7 r1 :| @p48
247:    v14r2..o518(cd),0& dffdq7|:(ga):|q8@M96rf ^1 ^1
248:
249:/ E.G.(R)
250:(t7)          @p80o518 @H32@M96
251:(t7)          (ef)&f2^16&@B0,-683f16@B0ed e(ef)&f4.&@M@B0,-2048f
@B0r>
252:(t7)          (a16b) &b8.b{ab}8<dedr@M96a 4aa2^&@M(ac)
253:/             **I**
254:    |:8 r1 :| r1
255:/             **A**
256:    |:16 r1 :|
257:/             **B**
258:    |:8 r1 :|
259:/             **C**
260:    |:7 r1 :| @p80
261:    v14r2..o418(ab)& b<dd>bq7|:(bb!+):|q8@M96ra ^1 ^1
262:
263:
264:/ Synthe.Hit
265:(t8)          o718rrr'c>c'r2 rrr'd>d'r2 r1 r1
266:/             **I**
267:    |:8 r1 :| r1
268:
269:/             **A**
270:    @u064|:|:r2rcrr:| r1 r1 |:r2rdrr:|
271:    |r1 rc>b-<c>a<c>g<c:|
272:     r1 r1
273:
274:/             **B**
275:    @92@u100 r1 @p64r1 o4a1 f+2.r4  r1 r1 'f1a'&'g1a'
276:
277:/             **C**
278:    |:8 r1 :|
279:    r1 r1 r1
280:
281:/ Chorus
282:(t9)          r1 r1 r1 r1 @u100
283:/             **I**
284:    o418|:|:4'a4<c':| 'a<d''a4<d'rr2
285:    ||:3'f4b<d':|'fb<d''g8^2<ce'r2 :|
286:     |:'b4<d':|'b<e''b4<e''a8^2.<f' r4 r1
287:
288:/             **A**
289:    r1 'c4gb'r'c2^fa' r1 r1
290:     r1 'ea<c'r|:'dgb'@D1:|r4.@D0r r1 r1
291:    r1 'cgb''cfa'r'c4.fa'rr r1 r1
292:     r1 |:'ea<c':||:'dgb'@D1:|r4.@D0r r1 r1
293:
294:/             **B**
295:    @u127'b1.^<d'r4. <(c1^d),192&c2^r4
296:     'd1>gb'& 'c2>gb'r2 (c1^e-)&c2^r4
297:
298:/             **C**
299:    u-16r2.'d4>b'|:'e>b':|r2. r1 r4'd>a''d4>a''d>a'r4
300:     r1 r4.'c4.>ae''c>ge'r r1 >r4|:4'ca':|r4
301:
302:
```

▶あの大地震が起こった当日，某放送局は「もし東京で起こったなら」とシミュレーションを始め，東京の某大学教授は「東京でなくてよかった」といい出す始末。まったく他人事のようである。不信感を抱いてしまった。
福知　健(23)京都府

```
303:/ Drums
304:/ SD,BD&Tom
305:(t10)         o218 |:crdcrrdr :| crdcrddc r4crbaad16d16
306:
307:/            **I**
308:    |:13 crdr :| crdc ru99|:7'gd'u+4:| 'gd'rrbragr
309:
310:/            **A**
311:    |:7 crrcrrdr :| crrcrdgf
312:    |:7 crrcrrdr :| ru111|:4gu+4:|dd16a16g
313:
314:/            **B**
315:    |:7 crdcrrdr :| rdrrdbag
316:
317:/            **C**
318:    |:7 crdcrrfr :| crdb16b16aagr
319:
320:    dddu-16d16d16udcrc
321:
322:/ Cymbals
323:(t11)        [K.Sign +f,+c,+g,+a] @u100
324:    o318c>u-32ffagfff ufu-32ffuau-32gfff
325:     ufu-32ffuau-32grrua gf<c2.
326:
327:/            **I**
328:    a!*0>|:13 ufu-32fff :| ufu-32ffu<c r1 r1
329:
330:/            **A**
331:    c*00>|:7 ufu-32ffuagagf :| fu-32ffuagrrr<
332:    a!*0>|:7 ufu-32ffuagu-32fff :| r1<
333:
334:/            **B**
335:    a!*0>|:7 ufu-32ffuagu-32fff :| fuagu-32fufrrr
336:
337:/            **C**
338:    <a!*0>|:7ufu-32ffufu-32ffff:| fu-32ffurr2
339:    r2rar<c r1 r1
340:
341:/ Tambourine&Cabasa
342:(t12)        [K.Sign +f] @u108 o314 |:5rf:|r.ff8fr2
343:    |:14rf:| r1 r1
344:
345:    o4116@u64 |:15 r8aaa2. :| r1
346:
347:    |:8 r1 :|
348:
349:    |:8 r1 :|
350:    r1 r1 r1
351:
352:*
353:*   YM-2151 Part
354:*
355:
356:/ぴよぴよ音はご勘弁を。
357:/ Right
358:(t13)        r1 r1 r1 r1
359:/            **I**
360:    @72o6112 Z95,98,101,103,106,106,103,103,101,98
361:    |:r4P2fa<cfa<c>{afc>a}4 r4P1fa<cdfa{fd>af}4
362:    | r4P2gb<dgb<d>{bgd>b}4 r4P1cgb<egb<c>bg>:|
363:      r4P2gb<d>r4P1{cegb}4 r1 r1
364:/            **A**
365:    |:r4P2cfa<cfa<{c>afc}4 >a2P1>cfa<cfa
366:      r4P2dfa<dfa<{d>afd}4> r4P1dfa<dfafd>a
367:      r4P2dgb<dgb<{d>bgd}4> r4P1dgb<dgb<{d>bgd}4>
368:      |r4P2|:cegb<cegbe >|r4P1:|:|
369:       r4P2cegb<cegbe P3Z101{ceg&}8g4. >{g<ceg}4b4
370:/            **B**
371:    |:8 r1 :|
372:/            **C**
373:    @71o618v11q4p2 ffffffff gggggggg aaaagggg |:8f+:|
374:    |:24b:| |:8a:| q8
375:
376:
377:/ Left
378:(t14)        r1 r1 r1 r1
379:/I-Aは頭と尻以外は(t13)と同じです。
380:             **I**
381:    @72o6112 Z92,95,98,100,103,103,101,101,99,96r32
382:    |:r4P2fa<cfa<c>{afc>a}4 r4P1fa<cdfa{fd>af}4
383:    | r4P2gb<dgb<d>{bgd>b}4 r4P1cgb<egb<c>bg>:|
384:      r4P2gb<d>r4P1{cegb}4 r1 r1
385:/            **A**
386:    |:r4P2cfa<cfa<{c>afc}4 >a2P1>cfa<cfa
387:      r4P2dfa<dfa<{d>afd}4> r4P1dfa<dfafd>a
388:      r4P2dgb<dgb<{d>bgd}4> r4P1dgb<dgb<{d>bgd}4>
389:      |r4P2|:cegb<cegbe >|r4P1:|:|
390:       r4P2cegb<cegbe P3Z98{ceg&}8g4. >{g<ceg}4b8.^32
391:/            **B**
392:    |:8 r1 :|
393:/            **C**
394:    @71o618v11q4p1 dddddddd eeeeeeee ffffeeee dddddddd
395:    ffffffff |:16e:| |:8f:| q8
396:
397:/ Center
398:(t15)        r1 r1 r1 r1
399:/            **I**
400:    @72o6112 Z93,94,96,98,99,98,97,96,95,94
401:    |:r4fa<cfa<c>{afc>a}4 r4fa<cdfa{fd>af}4
402:      |r4gb<dgb<d>{bgd>b}4 r4cgb<egb<c>bg>:|
403:       r4gb<d>r4{cegb}4 r1 r1
404:/            **A**
405:    |:r4cfa<cfa<{c>afc}4 >a4r4>cfa<cfa
406:       r4dfa<dfa<{d>afd}4> r4dfa<dfafd>a
407:      |:r4dgb<dgb<{d>bgd}4 :|>
408:      ||:r4cegb<cegbe:|>:|
409:         r4cegb<cegbe Zv10{ceg&}8g4. >{g<ceg}4b4
410:/            **B**
411:    |:8 r1 :|
412:/            **C**
413:    @71o518v11q4 bbbbbbbb<cccccccc ccccc+c+c+c+>aaaaaaaa
414:    <dddddddd |:24c:| q8
415:
416:/  このパートは ZMEX.x というプリプロセッサで入力しました。
417:/がむばって入力して下さい。
418:/  "q1'c8a'q8"は他の文字で入力しておいて("%"とか)、
419:/後でエディタの機能などを利用してまとめて変換すると楽です。
420:/
421:/ A.G.
422:(t16)        o314
423:/F       |Dm      |Gm7  C7|C7 ||
424:    'fa<cf',1q1'c8a'q8'fa<cf''f8a<cf'q1'c8a'q8r8
425:    'da<df'  q1'c8a'q8'da<df''d8a<df'q1'c8a'q8r8
426:    'fb<dg'  q1'c8a'q8'fb<dg''f8b<dg'q1'c8a'q8'cgb<ce'
427:    'c16gb<ce'r16'c2.gb<ce'
428:
429:/            **I**
430:/F       |Dm     ||Gm      |C7     :|Gm  C7 |F       |F      |
431:    14 |:'fa<cf'q1'c8a'q8'fa<cf''f8a<cf'q1'c8a'q8r8
432:         'da<df'q1'c8a'q8'da<df''d8a<df'q1'c8a'q8r8
433:       | 'fb<dg'q1'c8a'q8'fb<dg''f8b<dg'q1'c8a'q8r8
434:         'cgb<ce'q1'c8a'q8'cgb<ce''c8gb<ce'q1'c8a'q8r8 :|
435:         'fb<dg''f8b<dg'q1'c8a'q8|:'c8gb<ce':|q1'c8a'q8'fa<cf'
436:       18|:8'fa<cf':|r2..
437:
438:/            **A**
439:/|:F     |         |Dm      |        |
440:/Gm7     |         |C7      |        :|
441:    14 |:|: 'fa<cf'q1'c8a'q8'f.a<cf'q1'c8a'q8r8 :|
442:            |:'da<df'q1'c8a'q8'd.a<df'q1'c8a'q8r8 :|
443:          |: 'fb<dg'q1'c8a'q8'f.b<dg'q1'c8a'q8r8 :|
444:            |:'cgb<ce'q1'c8a'q8'cgb<ce'
445:                'c8gb<ce'q1'c8a'q8'c8gb<ce' :| :|
446:
447:/            **B**
448:/Gm7     |C      ||Am      |D      :|F      |F7      |
449:    |:'fb<dg'q1'c8a'q8'f.b<dg'q1'c8a'q8r8
450:      'ceg<ce'q1'c8a'q8'c.eg<ce'q1'c8a'q8r8
451:    | 'eg<ce'q1'c8a'q8'e.g<ce'q1'c8a'q8r8
452:       'da<df+'q1'c8a'q8'd.a<df+'q1'c8a'q8r8 :|
453:      'fa<cf'q1'c8a'q8'f.a<cf'q1'c8a'q8r8
454:       'e-2.a<cf',4 q1'c8a',1q8r8
455:
456:/            **C**
457:/Bb      |C         |F  Am  |D        |
458:/Gm7     |C         |C7     |F        |
459:    'fb<df'q1'c8a'q8'f.b<df'q1'c8a'q8r8
460:    'ceg<ce'q1'c8a'q8'c.eg<ce'q1'c8a'q8r8
461:    'fa<cf'q1'c8a'q8q1'c8a'q8'eg<c+e'q1'c8a'q8q1'c8a'q8
462:    'da<df+'q1'c8a'q8'd.a<df+'q1'c8a'q8q1'c8a'q8
463:    'fb<dg'q1'c8a'q8'f.b<dg'q1'c8a'q8r8
464:    'ceg<ce'q1'c8a'q8'c.eg<ce'q1'c8a'q8r8
465:    'cgb<ce'q1'c8a'q8'c.gb<ce'q1'c8a'q8r8
466:    'fa<cf'q1'c8a'q8'f.a<cf'q1'c8a'q8r8
467:/Gm7C7 F|F       |
468:    'fb<dg''f8b<dg'q1'c8a'q8|:'c8gb<ce':|q1'c8a'q8
469:    'fa8^1<cf'
470:
471:(p)
```

リスト2 ミンキーモモ用カウンタ表示

```
 1:00002430 00000000   2:00002430 00000000   3:00002430 00000000   4:00002430 00000000
 5:00002430 00000000   6:00002430 00000000   7:00002430 00000000   8:00002430 00000000
 9:000021F0 00000000  10:000022B0 00000000  11:00002430 00000000  12:00002430 00000000
13:000021F0 00000000  14:000021F0 00000000  15:000021F0 00000000  16:00002370 00000000
```

リスト3 ファイナルファンタジーIIメインテーマ

```
1:.comment [ Final Fantasy II] MainTheme Arrange Ver.  by LYRE
2:(i)(b0)
3:
4:.adpcm_block_data = ff2mtar.zpd
5:
6:/        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  口笛
7:(@ 1,    25, 17,  0,  9,  0, 37,  0,  8,  0,  0,  0
8:         25, 10,  4,  9,  0, 44,  0,  6,  1,  0,  0
9:         25, 10,  0,  9,  0, 54,  0,  2,  4,  0,  0
10:        13, 10,  4, 10,  0,  0,  0,  2,  0,  0,  0
11:/       AL  FB  OM
12:         0,  7, 15)
13:
14:/       AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Strings
15:(@ 2,   18, 16,  0, 10,  0, 33,  0,  2,  3,  0,  0
16:        18, 16,  0, 10,  0,  5,  0,  1,  7,  0,  0
```

▶「ANOTHER CG WORLD」のグラサンの人が女性だと初めて知りました。申しわけない。
長野　大介(20)福岡県

```
17:         18, 16,  0, 10,  1, 29,  0,  2,  7,  0,  0
18:         27, 16,  0, 10,  0,  4,  0,  2,  3,  0,  0
19:/        AL  FB  OM
20:          4,  7, 15)
21:
22:/        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Bass
23:(@ 3,    22,  8,  3,  4,  2, 32,  1,  3,  3,  0,  0
24:         22,  0,  8, 10,  0,  0,  0,  1,  7,  0,  0
25:         22,  0,  3,  2,  1, 15,  1,  3,  7,  0,  0
26:         22,  0,  8, 10,  2, 12,  0,  2,  3,  0,  0
27:/        AL  FB  OM
28:          4,  7, 15)
29:
30:/        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Harp
31:(@ 4,    25, 16, 12,  8,  0, 35,  0,  2,  0,  0,  0
32:         25, 16, 12,  8,  0,  5,  0,  1,  0,  0,  0
33:         25, 16, 12,  8,  0, 26,  0,  2,  0,  0,  0
34:         25, 16, 12,  8,  0,  0,  0,  1,  0,  0,  0
35:/        AL  FB  OM
36:          4,  7, 15)
37:
38:/        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Horn
39:(@ 5,    13, 12,  0,  9,  0, 35,  0,  2,  2,  0,  0
40:         16, 12,  2, 12,  0, 10,  0,  4,  6,  0,  0
41:         16, 12,  2, 12,  0, 10,  0,  4,  6,  0,  0
42:         16, 12,  2, 12,  0, 10,  0,  4,  6,  0,  0
43:/        AL  FB  OM
44:          5,  7, 15)
45:
46:/        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  PSG I
47:(@ 6,    31,  2,  2, 15,  0, 27,  0,  2,  3,  0,  0
48:         18,  4,  4, 15,  0, 10,  0,  1,  7,  0,  0
49:         31,  2,  2, 15,  0, 48,  0,  4,  7,  0,  0
50:         18,  4,  4, 15,  0, 15,  0,  1,  3,  0,  0
51:/        AL  FB  OM
52:          4,  7, 15)
53:
54:/        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  PSG II
55:(@ 7,    31,  0,  0, 15,  0, 27,  0,  2,  0,  0,  0
56:         31,  0,  0, 15,  0, 50,  0,  1,  0,  0,  0
57:         31,  0,  0, 15,  0, 40,  0,  2,  0,  0,  0
58:         31,  0, 12, 15,  0,  4,  0,  1,  0,  0,  0
59:/        AL  FB  OM
60:          1,  7, 15)
61:
62:(m1,1000)(m2,1000)(m3,1000)(m4,1000)(m5,1000)(m6,1000)
63:(m7,1600)(m8,1600)(m9,100)(m10,200)(m11,100)(m12,600)
64:(m13,600)(m14,600)
65:(aFm1,1)(aFm2,2)(aFm3,3)(aFm4,4)(aFm5,5)(aFm6,6)
66:(aFm7,7)(aFm8,8)(aAdpcm,9)(aAdpcm,10)(aAdpcm,11)
67:(aAdpcm,12)(aAdpcm,13)(aAdpcm,14)
68:
69:
70:/------------------------------------------------------------
71:/         口笛 & Horn
72:/------------------------------------------------------------
73:(t1)
74:t102
75:o4 @6 v12 @q1 p3 18 K1 @k0  r8
76:(t2)
77:o4 @6 v9 @q1 p2 18 K1 @k10 r8
78:r16..
79:
80:(t1,2)
81:>a*7b*7<c*7d*7e*7f+*7g+*6
82:|:a2.<e4>b2.e4 <c4>b4a4b4g2e2
83:|1 f2^fa<ce4d4g4f4 d+2.>b4<e2&t99e&t96e&t93e&t90e>t102
84::|
85:f2^fga<c4>b-4a4g+4
86:
87:(t1)
88:t100
89:a&t98a&t96at94b<t92c&t90c&t88ct86dt84e&t82e&t80e&t78e&t60e4>
90:t135
91:o4 @1 v15 @q2 p3
92:@s6 @h40 @m20
93:
94:(t2)
95:a4.b<c4.de2.>
96:o4 @1 v13 @q2 p2
97:@s6 @h40 @m20
98:r16..
99:
100:(t1,2)
101:[do]
102:>a*7b*7<c*7d*7e*7f+*7g+*6
103:a2.<e4>b2.e4 <c4>b4a4b4g2e2 f2^fa<ce4d4g4f4 d+2.>b4<e1>
104:a2.<e4>b2.e4 <c4>b4a4b4g2e2 f2^fga<c4>b-4a4g+4 a1
105:
106:(t1)
107:r1
108:@514o3p2@m0v13@k8
109:rfafg2f2&_15f~15egea2g2&_15g~15faf&_15f~15f+af+g+*240&_15g+2
110:@118o4p3@m20v15@k0
111:[loop]
112:
113:(t2)
114:r2.r32
115:@514o3p1@m0v11@k7
116:rcfcd2d2&_15d~15>b<e>b<e2e2&_15e~15cfc&_15c~15d+f+d+e*240&_15e
2
117:@118o4p2@m20v13@k10r8..
118:[loop]
119:
120:
121:/------------------------------------------------------------
122:/         String
123:/------------------------------------------------------------
124:(t3)
125:o4 @2 v12 @q2 p3 18 K1 @k6 r8
126:@s10 @m5
127:
128:(t12)
129:o4 v9 p3 18 K1 @f3 r8
130:
131:(t3,12)
132:r4
133:|:4 r1r1r1r1 :|
134:[do]
135:e1g1a2f2e4>b<dc+4de f1a2g2b2a2g+2d2
136:e1g1a2f2e4>b<dc+4de f1e-2e2e1 ~5a4b4<c4d4>
137:<e1d2^c>b<cd2.>b4<d2c4>ab
138:<c1>b2a2<e1&_6e&_6e~7>e4f+4g+4
139:[loop]
140:
141:
142:(t4)
143:o4 @2 v11 @q2 p3 18 K1 @k4 r8
144:@s10 @m5
145:
146:(t13)
147:o4 v9 p3 18 K1 @f3 r8
148:
149:(t4,13)
150:r4
151:|:4 r1r1r1r1 :|
152:[do]
153:c1d1f2c2>b2a4b<c+ d1f2e-2f+2e-2e2>b2<
154:c1d1f2c2>b2a4b<c+ d1>b-2b2<c1 r1
155:~5<c1>b2.g4b2.g4a2a4g4
156:a1a2f+2b1&_6b&_6b~7>b4<d4d4
157:[loop]
158:
159:
160:(t5)
161:o3 @2 v12 @q2 p1 18 K1 @k5 r8
162:@s10 @m5
163:
164:(t14)
165:o3 v9 p3 18 K1 @f3 r8
166:
167:(t5,14)
168:r4
169:|:4 r1r1r1r1 :|
170:[do]
171:a1b1<c2>a2g2e2 a1<d2>b-2<e-2>b2b2g+2
172:a1b1<c2>a2g2e2 a1g2g+2a1 ~5a4b4<c4d4>
173:<a1g2.d4g2.e4e2e4e4>
174:<f1f+2d+2g+1&_6g+&_6g+~7>g+4a4b4
175:[loop]
176:
177:
178:/------------------------------------------------------------
179:/         Bass
180:/------------------------------------------------------------
181:(t6)
182:o1 @3 v14 q8 p3 18 K1 @k0 r8
183:r4
184:|:4 r1r1r1r1 :|
185:[do]
186:|:aa&_15a2~15<ae>gg&_15g2~15<g8>&(<g8,>g)
187:   ff&_15f2~15<fc>ee&_15e4~15aag4
188:|1 dd&_15d2~15<d>ab-b-&_15b-4~15b-b-&_15b-4~15
189:   bb&_15b2~15<e->e-ee&_15e4~15g+g+<e8>&(<e8>,e)
190::|
191:dd&_15d2~15<d>ae-e-&_15e-2~15<e8>&(<e8,>e)
192:aa&_15a2~15<ae>a<a>b<bc<c>e<e>>
193:
194:ff&_15f2^~15fgg&_15g2~15gf
195:ee&_15e2~15<e>baa&_15a2~15g4
196:dd&_15d2~15<d>ad+d+&_15d+2~15<d+>b
197:ee&_15e2~15<e>be8&(e8,<e>)<e>f+4<f+>g+<g+>
198:[loop]
199:
200:
201:/------------------------------------------------------------
202:/         Harp
203:/------------------------------------------------------------
204:(t7)
205:o4 @7 v12 q6 p2 116 K1 @k10 r8
206:r418p3
207:
208:(t8)
209:o4 @7 v7  q6 p3 116 K1 @k12 r8r16
210:r418p1
211:
212:(t7,8)
213:|:|:>a<efe:||:>g<efe:||:>f<ded:|>e<ded>e<c+dc+
214:|1 >db<ce>ca<ce>>b<b<df>>b-<<cdf
215:   >d+ab-<d+>>b<f+b<d+>eab<ed>bg+e< :|
216:>db<ce>ca<ce>>b-<a<df>>b<ab<e>ab<ef>fab<f>eab<de2
217:@4116
218:
219:(t7) p2v13
220:(t8) p3v8 r16
221:
222:(t7,8)
223:[do]
224:|:acea<c>ea<ce>e<aec>aec g>b<dgbdgb<d>d<gd>bgd>b<
225:fcfa<c>fa<cf>f<afc>afc e>b<egbegb<c+>e<ec+>aec+>a<
226:d>a<dfadfa<d>a<fd>afd>a< |1 d>a<dfadfa<e->b-<ge->b-ge->b-<
227:e->b<e-g-be-g-b<e->g-<bg-e->bg-e-e>b<eg+beg+b<e>b<g+ed>bg+e
228::|
229:e->b-<e-gb->b-<e-gbe<e>bg+ed>b<
230:acea<c>ea<ce>e<aec>aec acea<c>ea<ce>e<aec>aec
231:
232:>fa<cefa<cefec>afec>a egb<degb<ded>bged>bg<
233:>egb<degb<ded>bged>bg eab<ceab<cec>baec>ba<
234:>fa<cefa<ceaec>afec>a f+ab<d+f+ab<d+f+d+>baf+d+>ba<
235:>eab<deab<ded>baed>ba eab<deab<e>f+ab<d>g+b<de>
236:[loop]
237:
```

▶ああ，夜景がキレイだ。赤，白，緑，青，黄，オレンジ。さあ，あなたも機器のスイッチを全部入れて部屋の明かりを消そう。LEDの夜景が浮かび上がるぞ。

渡辺　久孝(27)大阪府

```
238:
239:
240:/------------------------------------------------------------
241:/       Adpcm Drum
242:/------------------------------------------------------------
243:(t9)
244:o2 v9 p3 l8 k0 r8
245:r4
246:|:4 r1r1r1r1 :|
247:[do]
248:ccr2rcc4c4r2
249:[loop]
250:
251:
252:(t10)
253:o2 v9 p3 l16 k1 r8
```

```
254:r4
255:|:4 r1r1r1r1 :|
256:[do]
257:@v71a8_7f_7f_7f_7f_7f_7f_7a8~8f~8f~10a8~8f~8f
258:[loop]
259:
260:(t11)
261:o2 v9 p3 l8 k0 r8
262:r4@r1
263:|:4 r1r1r1r1 :|
264:[do]
265:r2e2r2e2
266:[loop]
267:
268:
269:(p)
```

リスト4　FF2用コンフィグファイル

```
 1: 0 = slams.pcm, v30
 2: 1 = slams.pcm, v40, m0, d1800
 3: 2 = slams.pcm, v70, m1, d1800
 4: .o2c  = snapk.pcm
 5: .o2e  = slams.pcm, m2, d1800
 6: .o2f+ = ch1.pcm
 7: .o2a+ = oh1.pcm
 8:
 9: .o3f  = str_c5.pcm, p-12
10: .o3f+ = str_c5.pcm, p-11
11: .o3g  = str_c5.pcm, p-10
12: .o3g+ = str_c5.pcm, p-9
13: .o3a  = str_c5.pcm, p-8
14: .o3a+ = str_c5.pcm, p-7
15: .o3b  = str_c5.pcm, p-6
16: .o4c  = str_c5.pcm, p-5
17: .o4c+ = str_c5.pcm, p-4
18: .o4d  = str_c5.pcm, p-3
19: .o4d+ = str_c5.pcm, p-2
```

```
20: .o4e  = str_c5.pcm, p-1
21: .o4f  = str_c5.pcm
22: .o4f+ = str_c5.pcm, p1
23: .o4g  = str_c5.pcm, p2
24: .o4g+ = str_c5.pcm, p3
25: .o4a  = str_c5.pcm, p4
26: .o4a+ = str_c5.pcm, p5
27: .o4b  = str_c5.pcm, p6
28: .o5c  = str_c5.pcm, p7
29: .55c+ = str_c5.pcm, p8
30: .o5d  = str_c5.pcm, p9
31: .o5d+ = str_c5.pcm, p10
32: .o5e  = str_c5.pcm, p11
33: .o5f  = str_c5.pcm, p12
34:
35: .erase 0
36: .erase 1
37: .erase 2
```

リスト5　FF2用PCM作成バッチファイル

```
zplk violae4lp.p16 -f8000,60,1 str_rel.p16
zplk -x0,6 violae4lp.p16 str_lp.p16
zplk violae4at.p16 -x0,1 str_lp.p16 str_al.p16
zplk str_al.p16 -x0,1 str_rel.p16 -T41124,65280 -v20 -a str_c5.pcm
zpcnv FF2MTAR.CNF
```

```
del str_rel.p16
del str_lp.p16
del str_al.p16
del str_c5.pcm
```

リスト6　FF2用カウンタ表示

```
 1:00000C18 00001200   2:00000C42 00001200   3:00000C48 00001200   4:00000C48 00001200
 5:00000C48 00001200   6:00000C48 00001200   7:00000C48 00001200   8:00000C60 00001200
 9:00000C48 00000180  10:00000C48 000000C0  11:00000C48 00000180  12:00000C48 00001200
13:00000C48 00001200  14:00000C48 00001200
```

リスト7　別れの曲

```
 1:.COMMENT      É t u d e(別れの曲) F.Chopin   By T.Takahashi
 2:
 3:(I)
 4:(B1)
 5:
 6:(m1,3000)(aMIDI1,1)
 7:(m2,3000)(aMIDI1,2)
 8:(m3,3000)(aMIDI1,3)
 9:(m4,3000)(aMIDI1,4)
10:(m5,3000)(aMIDI1,5)
11:(m6,3000)(aMIDI1,6)
12:
13:/-------------------------------------------------------------
14:/                         S E T  U P
15:/-------------------------------------------------------------
16:
17:.ROLAND_EXCLUSIVE $10,$42={ $40,$00,$7F,$00 }
18:.SC55_V_RESERVE $10={ 24,0,0,0,0,0,0,0,0,0,0,0,0,0,0 }
19:
20:(t1,2,3,4,5)
21:   @I$41,$10,$42 @E100 @p64
22:   Y$42,96                          / Sostenute
23:
24:/-------------------------------------------------------------
25:/                           M M L
26:/-------------------------------------------------------------
27:
28:(t1,2,3,4,5)
29:   [K.SIGN +c,+d,+f,+g]
30:   I0 @1 q8 v16 r32
31:
32:/ Piano 1
33:(t1)
34:/---1.1
35:   @u56o3b8 <e8@u60L16d@u62ef4& fz64,66,64,70ggfg4&
36:   gz64,66,64,aag<c8.@u58>b agde@u66f4&
37:/---1.2
38:   fz68,70,68,74ggfe4 z66,68,70,72,74,76,78,80gafgabga
39:   z82,90,82,88,80<c8>f4a+*2g*10f& f8@u56a+*2g*10f@u48b4
40:   z56,58,60,64g8def4&
41:/---1.3
42:   fz66,68ggfg4& gz66,68,66,60,56aag<c8.>b @u52agde@u58f4&
43:   fz60,62,64,58ggfe4 z58,,62,b<cc>b@u66abga
44:/---1.4
45:   <deedcz78,82,86d>b+<c
46:   q7z64,70,76,82,88,92,96,100efdefgef
47:   q8z88,84,80,76g4^fec @u72d4edc>g
48:/---1.5
49:   b4<cz68,64,60>bae @u56g2& z40,56,54,52,50e4<ede>b
50:   @u56<dcd>gba+@u52<c8
51:   @u48>a+*2g*10fa+*2g*10fz56,60,64,68<ede>b
52:/---2.1
53:   @u72<dcd>gba+@u68<c8
54:   z60,58,56>a+*2g*10fa+*2g*10fz88,86,84,82<fefc
55:   @u84ede>a+<c>b+@u80<d8
56:   z80,76,80,,76,80>b*2a*10gb*2a*10gz88,86,84,82<fefc
57:   @u84ede>a+<c>b+@u80<d8
58:/---2.2
59:   z56,54,,52,50,>b*2a*10gb*2a*10g
60:   z60,62,70,78,86b*2a*10gab @u88<c8.c>b8.@u56b
61:   <c!8.c!>b8.b q5z108,56,58,64,66,64,78,86<cq8e>b<cg!beg!
62:/---2.3
63:   z90,98,102,110,114<ce>b<cg!q7z108,104,96>g!fe
64:   q8@u88d8.dc8.@u56c d!8.d!c8.c
65:   q5z108,56,58,64,66,68,78,86dq8fcda<c>fa
66:/---2.4
67:   z100,98,102,110<dfcd@u114aq7>az110,106gf
68:   q7z102,72,80,88,96,104,112,120e+q8d!cfe+a+a<d!
69:   q7cq8>cc!f!eag<c
70:/---2.5
71:   q7z102,72,76,80,84,88,92,96c!q8>c!>b<edgg!<c!
72:   z100,104,108,112,116,120,124>b<edgg!<c!>b@u127<e
73:   q7dq8>bfr8>>bfr
74:/---3.1
75:   rq6az120,124b<c!@u127>bab<c! q5>bq8<<<d>br8>d>br
76:   rq6ab<c!>bab<f
```

▶小学校の体育館で書いています。1月19日，午後11時25分。そう，あの兵庫県南部地震のため避難しています。電気のみの生活ってつらすぎます。パソコンのデータが大事なのでチェックしてフロッピーディスクに転送する予定です。同じ兵庫県の人たち，ともにがんばりましょう。　菅谷　英明(28)兵庫県

```
77:/---3.2
78:    q5fq8<bf!d!gadc! fadd!gg!cc! fadc!fgd!>b
79:/---3.3
80:    <f!gd!cg!fc!>b <f!gd!>b<f!fc!>a <dfc!>b<f!e>a+a
81:/---3.4
82:    <dd!>gg!<c>g!<c>g!
83:    <c!>fz122,116,110,104,98,92<c!>f<c!>f<c!>f
84:    @u88br@u56>bgba@u44f8&
85:/---3.5
86:    f8@u56bgba@u64f8& f8@u56gffe@u64c8
87:    z56,,64edg@u56ffe@u64c8 z56,66,56,52,56,40ed8ced8c
88:/---4.1
89:    z36,44,36ed8ced8c @u36edededed ededed c8d8 @u56e8def4&
90:/---4.2
91:    fz58,60gg@u64fg4& gz58,60,62,48aag<c8.@u52>b
92:    agde@u46f4& fz48,50,52,48ggfe4
93:/---4.3
94:    z46,50,54,58,56,58,60,62b<cc>babga
95:    z64,68,72,74,78,80,82,84<deedcd>b+<c
96:    q7z64,70,76,82,88,92,96,100efdefgef
97:/---4.4
98:    q8z88,,66,62,58g4&gfec @u56d4edc>g
99:    b4z52,48,64@u40<c>bae g4&gz48,44,40fec!
100:/---4.5
101:    @u36g4&gfec! g2& g4&gq7L8fe>b q8<e1
102:
103:/ Piano 2
104:(t2)
105:/---1.1
106:    o4r8 L4r@u62d& d8z66,70d8e r2 r@u46L16cdcd
107:/---1.2
108:    cdcdr4 r2 r2 r4@u48d4 z56,64e8r8d4&
109:/---1.3
110:    d8@u68d8e4 r2 r4@u48cdcd cdcdr4 z58,60,62,64fgfgr4
111:/---1.4
112:    @u66b+ra+b+r4 q7z64,70,76,82,88,92,96,100
113:    a+a+a+a+a+a+a+a+ q8@u88<e4r4 r2
114:/---1.5
115:    r2 r4.@u40>gf r2 r2 r2
116:/---2.1
117:    r2 r2 r2 r2 r2
118:/---2.2
119:    r2 rz86,78>ee8.ee8 r@u56ee8.ee8 q5@u108<er4..
120:/---2.3
121:    r4^q7z108,104,96eee q8z88,96>bff8.ff8
122:    r@u56ff8.ff8 q5@u108<fr4..
123:/---2.4
124:    r4^q7z114,110,106fff q6@u102e+r4.. r2
125:/---2.5
126:    r2 r2 r2
127:/---3.1
128:    rq6fz120,124ff@u127ffff q5fr4.. rq6ffffff<e
129:/---3.2
130:    q5dr4.. r2 r2
131:/---3.3
132:    r2 r2 r2
133:/---3.4
134:    r2 r2 r2
135:/---3.5
136:    r2 r2 r2 r4.q8@u44>>g!8
137:/---4.1
138:    z36,44f8r4g!8 @u36f8r4. r2^8 r4@u44<d4&
139:/---4.2
140:    d8d8e4 r2 r4cdcd cdcdr4
141:/---4.3
142:    z46,50,54,58fgfgr4 z64,72,74b+ra+b+r4
143:    q7z64,70,76,82,88,92,96,100a+a+a+a+a+a+a+a+
144:/---4.4
145:    q8@u88<e4r4 r2 r2 r2
146:/---4.5
147:    r2 r2 r2^8. r1
148:
149:/ Piano 3
150:(t3)
151:/---1.1
152:    o3r8 @u48L16gbgbabab ababgbgb
153:    g<e>b<e@u46da>b<d >b<e>gbabab
154:/---1.2
155:    ababgbgb <d!z56,58,60,64,66,68,70ed!ecece
156:    @u56cece>b<e>b<e >a+<e>a+<e@u48>abab
157:    z50,52,54gbg@u52babab
158:/---1.3
159:    abab@u58gbgb @u56g<e@u52>b<eda>b<d @u48>b<e>gbabab
160:    ababgbgb z58,60,62,64<d!ed!e@u66cece
161:/---1.4
162:    fgfgez76,80,84geg q7z64,70,76,82,88,92,96,100efdefgef
163:    q8@u52gbgbgz50,48,46bgb @u44dfdfeger
164:/---1.5
165:    >b<d>b<dcz46,44,42ecr @u40>gbgb@u36gbgb
166:    g4z56,54,52,50<gfgd @u56fefde8@u52gf
167:    @u48ededz56,60,64,68gfgd
168:/---2.1
169:    @u72fefde8@u68gf @u64ededz88,86,84,82agae
170:    @u84gfgef8@u80ag fefez88,86,84,82agae @u84gfgef8@u80ag
171:/---2.2
172:    z54,,50,,62,70,78,86fefefefg @u88a8.ag8.@u56g
173:    a8.ag8.g q5z108,56,58,64,66,74,78,86bq8<c>g!b<eg!ce
174:/---2.3
175:    z90,98,102,110,114,108,104,96b<c>g!b<eq7>>a+a+a+
176:    q8@u88b8.ba+8.@u56a+ b8.ba+8.a+
177:    q5z108,56,58,64,66,74,78,86<cq8d>a<cfadf
178:/---2.4
179:    z90,98,102,110,114,,110,106<cd>a<cfq7>>b+b+b+
180:    q6z102,72,80,88,96,104,112,120bq8gg!<c!>b<edg
181:    q7g!q8>g!fba+<dd!g!
182:/---2.5
183:    q7z102,72,76,80,84,88,92,96fq8>ff!a+a<d!cf
184:    z100,104,108,112,116,120,124f!a+a<d!cfe+@u127a+
185:    q7bq8d>ar8>d>ar
186:/---3.1
187:    rq6<ez120,124ee@u127eeee q5dq8<adr8>adr rq6eeeeee<c!
188:/---3.2
189:    q5>bq8<<d!>gf!b<c!>fd a<c!>ff!ba+ed a<c!>fdabf!d!
190:/---3.3
191:    gaf!ea+add! gbf!d!gadc! fadd!gg!cc!
192:/---3.4
193:    ff!>ba+<e>a+<e>a+ <ez127,122,116,110,104,98,92
194:    >a+<e>a+<e>a<e>a @u88<dr@u56>d!ece@u44>a<e
195:/---3.5
196:    >a<dd!ece>a<e >a<d>abgbg!b
197:    fbabgbg!b fbz64,56,52,48,52,40g!bfbrb
198:/---4.1
199:    rz36,44bg!@u36bfbrb rbfbf<c>f<c >fafafbf8b8 @u44gbgbabab
200:/---4.2
201:    ababgbgb g<e>b<eda>b<d >b<e>gbabab ababgbgb
202:/---4.3
203:    z46,50,54,58,56,58,60,62<d!ed!ecece
204:    z64,68,72,74,78,80,82,84fgfgegeg
205:    q7z64,70,76,82,88,92,96,100efdefgef
206:/---4.4
207:    q8@u48gbgbgz46,44,42bgb @u40dfdfeger
208:    >b<d>b<dcecr @u36>gbgba<c!>ar
209:/---4.5
210:    @u32gbgba<c!>ar @u28gbgbgbgb gbgbgL8bgr g1
211:
212:/ Piano 4
213:(t4)
214:/---1.1
215:    o2r8 z64,56L16rb8brb8b rb8brb8b @u56rb8brb8b rb8brb8b
216:/---1.2
217:    rb8brb8b rz58,62,66,70<e8ere8e rz64,56f8frf8f
218:    @u56rf8frz46,44f8f rz58,62>b8br@u64b8b
219:/---1.3
220:    rb8br@u58b8b r@u56b8br@u52b8b rb8br@u48b8b
221:    rb8brb8b rz66,64,62<e8ere8@u68e
222:/---1.4
223:    rg8grz78,86g8g z64,70,76,82,88,92,96,100
224:    'fa+''fa+''fa+''fa+''fa+''fa+''fa+''fa+'
225:    @u72'g2b' @u48fb>b<bcg>g<g
226:/---1.5
227:    dg>g<g>az46,44,42<e>e<e @u40>b<e>e<e@u36>b<er>b&
228:    'b4<e'r4 q5@u48f<a+>f<b>fq8<<cz46,44>ba+
229:    z48,46,44>b<fbrr4
230:/---2.1
231:    q5@u56>f<a+>f<b>fq8<<cz54,52>ba+ >b<fbrr4
232:    q5@u84>g<b+>g<<c>>gq8<<dz82,80c>b+
233:    z84,82,80cg<crr4 q5@u84>>g<b+>g<<c>>gq8<<dc>b+
234:/---2.2
235:    z54,,50,,62,70,78,86cg<cq6>c>b<<d!>e>b
236:    rq8z86,78<<'ce''d!8.f''d!f''d!8e'
237:    r@u56'c!e''d!8.f!''d!f!''d!8e'
238:    q5z108,56,58,64,66,74,78,86>'g!<c'q8b<eg!ceb<c
239:/---2.3
240:    z90,98,102,110,114,108,104,96>g!b<eg!cq7>ccc
241:    rq8z96,88'df''e8.g''eg''e8f'
242:    r@u56'd!f''e8.g!''eg!''e8f'
243:    q5z108,56,58,64,66,74,78,86'd>a'q8cfadf<cd
244:/---2.4
245:    z90,98,102,110,114,,110,106>a<cfadq7>ddd
246:    q6z102,72,80,88,96,104,112,120d!q8>ba+agg!ff!
247:    q7eq8a+agg!ff!e
248:/---2.5
249:    q7z102,72,76,80,84,88,92,96dq8agg!ff!ed
250:    z100,104,108,112,116,120,124d!cc!>ba+ag@u127g!
251:    q7fr8q8<<<bfr8>>b
252:/---3.1
253:    fq6'c!a'z120,124>'b<b''a<<c!'
254:    @u127'b<b'<'c!a'>'b<b''a<<c!' q5'b<b'r8q8<<<<d>ar8>d
255:    >aq6'c!a'>'b<b''a<<c!''b<b'<'c!a'>'b<b''a<<f'
256:/---3.2
257:    q5'b<<f'q8<<b<f!d!>ga<dc!
258:    >fa<dd!>gg!<cc! >fa<dc!>fg<d!>b
259:/---3.3
260:    f!g<d!c>g!f<c!>b f!g<d!>bf!f<c!>a df<c!>bf!ea+a
261:/---3.4
262:    dd!gg!cg!cg! c!z127,122,116,110,104,98,92fc!fc!fc!f
263:    z88,56,54,64>brL8e>a<c!
264:/---3.5
265:    z54,64,,58>b<{efe}q7>aq8<c! z54,56,54,64>b>b<eb&
266:    b>{b<c>b}q7<eq8b& @u56b{efe}q7>bq8@u52<b&
267:/---4.1
268:    bz36,34,32,30{efe}q7br8 r4.q8{fgf}
269:    >b4r4^8 L16r@u44<b8brb8b
270:/---4.2
271:    rb8brb8b rz58,62,62,58b8brb8b r@u44b8brb8b rb8brb8b
272:/---4.3
273:    r@u48<e8ere8e r@u58g8grg8g z64,70,76,82,88,92,96,100
274:    'fa+''fa+''fa+''fa+''fa+''fa+''fa+''fa+'
275:/---4.4
276:    @u48'g2b' @u40fb>b<bcg>g<g
277:    dg>g<g>a<e>e<e @u36>b<e>e<ec!e>e<e
278:/---4.5
279:    @u32>b<e>e<ec!e>e<e @u28>b<e>e<e>b<e>e<e
280:    >b<e>e<e>bL8<e>eb& b1
281:
282:/ Piano 5
283:(t5)
284:/---1.1
285:    o2r8 @u56L4e>b b<e e>b <e>b
286:/---1.2
287:    b<e e@u64a @u56ab <c@u48>b e@u64>b
288:/---1.3
289:    b@u58<e z56,52e>b <e@u48>b b<e z58,68ea
290:/---1.4
291:    z74,78g<c z64,70,76,82,88,92,96,100
292:    L16'ce''ce''ce''ce''c!e''c!e''c!e''c!e' @u72>'b2<e' r2
293:/---1.5
294:    r2 L4r.@u36e8& er r2 r2
295:/---2.1
296:    r2 r2 r2 r2 r2
297:/---2.2
298:    r2 @u88a8.L16eb8.@u56e a8.eb8.e
```


▶「猫とコンピュータ」が100回目になるのですね。おめでとうございます。今後も連載を続けてほしいです。
直江　浩永(34)大阪府

```
299:    q5z108,56,58,64,66,74,78,86g!q8<g!<ce>b<cg!b
300:/---2.3
301:    z90,98,102,110,114,108,104,96eg!<ce>bq7>fff
302:    r8.q8@u88>f<c8.@u56>f b8.f<c8.>f
303:    q5z108,56,58,64,66,74,78,86aq8<a<dfcda<c
304:/---2.4
305:    z90,98,102,110,114,,110,106>fa<dfcq7>>ggg
306:    q6z102,72,80,88,96,104,112,120gq8f!edd!cc!>b
307:    q7a+q8<edd!cc!>ba+
308:/---2.5
309:    q7z102,72,76,80,84,88,92,96aq8<dd!cc!>ba+a
310:    z100,104,108,112,116,120,124gg!ff!edd!@u127c
311:    q7>br8q8<<<<d>ar8>d
312:/---3.1
313:    >aq6c!z120,124>ba@u127b<c!>ba
314:    q5br8q8<<<<f>br8>f >bq6c!>bab<c!>ba
315:/---3.2
316:    q5bq8<<<d!gf!>b<c!fd >a<c!ff!>ba+<ed >a<c!fd>ab<f!d!
317:/---3.3
318:    >gb<f!e>a+a<dd! >gb<f!d!>ga<dc! >fa<dd!>gg!<cc!
319:/---3.4
320:    >ff!ba+ea+ea+ ez127,122,116,110,104,98,92a+ea+eaea
321:    @u88>br4..
322:/---3.5
323:    r2 r2 r2 r2
321:/---4.1
325:    r2 r2 r2^8 L4@u44e>b
326:/---4.2
327:    b<e e>b <e>b b<e
328:/---4.3
329:    @u48ea @u58g<c z64,70,76,82,88,92,96,100
330:    L16'ce''ce''ce''ce''c!e''c!e''c!e''c!e'
331:/---4.4
332:    @u48>'b2<e2' r2 r2 r2
333:/---4.5
334:    r2 r2 r2^8. @u28e1
335:
336:/ Dumper & Tempo
337:(t6)
338:    r32
339:/---1.1
340:    L16t30r8 t20@D1rrt30@D0rrt25@D1rrrr
341:    t20r@D0@D1rt25@D0rr@D1rrrr t20rt25r@D0rr@D1rrrr
342:    @D0@D1rr@D0rr@D1rr@D0rr
343:/---1.2
344:    t20@D1rrr@D0rt25@D1rrrr
345:    t20@D0@D1rt23rt26rt29rt32@D0@D1rt35rt38rt30r
346:    t20@D0@D1rrt30rr@D0@D1rrr@D0r t25@D1rrt20r@D0r@D1rrrr
347:    t25@D0@D1rr@D0rr@D1rrrr
348:/---1.3
349:    t20@D0@D1rt25rr@D0r@D1rrrr t20rrr@D0r@D1t25rrrr
350:    @D0@D1rrr@D0rt20@D1rr@D0rr @D1rt23r@D0rt25r@D1rrrr
351:    t20@D0@D1rt23rt26rt29rt32@D0@D1rt35rt29rr
352:/---1.4
353:    t32@D0@D1rt35rt38rt41r@D0@D1rrt32rr
354:    t26@D0@D1rt32rt38rt44r@D0@D1t38rt32rt26rt20r
355:    @D0@D1rrrrrrt23rt26r t20@D0@D1rrrrt23@D0@D1rrr@D0r
356:/---1.5
357:    @D1rrrr@D0@D1rr@D0rr t20@D1rrrrrrrr
358:    @D0@D1rrrrt26@D0rt32rt38rt44r
359:    t50@D1rrrr@D0@D1rr@D0t47rt44r
360:    t32@D1rrr@D0rt38rt44rt50rt56r
361:/---2.1
362:    @D1rrrr@D0@D1rrt47@D0rt44r t38@D1rr@D0rrrt44rt50rt56r
363:    t62@D1rrrr@D0@D1rrt47@D0rt44r t38@D1rrr@D0rrt44rt50rt56r
364:    t62@D1rrrr@D0@D1rrt50@D0rt47r
365:/---2.2
366:    t38@D1rrt50@D0rrt56@D1rrt48@D0rt38r
367:    t56@D1rrrr@D0@D1rrr@D0r @D1rt48rrr@D0@D1rrt44r@D0t32r
368:    t20@D1rt80rrt100rrrrr
369:/---2.3
370:    rrrt90rt80@D0rt56rrt50r t56@D1rrrr@D0@D1rrr@D0r
371:    @D1rt48rrr@D0@D1rrr@D0r t20@D1rt80rrt100rrrrr
372:/---2.4
373:    rrrt80rt50@D0rt70rrr t80@D1rt90rrr@D0@D1rr@D0@D1rt70r
374:    t60@D0@D1rt90rrr@D0@D1rr@D0@D1rt70r
375:/---2.5
376:    t60@D0@D1rt90rrr@D0@D1rr@D0@D1rr
377:    @D0@D1rr@D0@D1rrt80@D0@D1rt70rt60@D0@D1rt50r
378:    t30@D0@D1rt95rrrrrrr
379:/---3.1
380:    @D0rt80rrt70rrt60rrt40r t20@D1rt95rrrrrrr
381:    t70@D0rrrt60rrt50rt40rt25r
382:/---3.2
383:    t20@D1rt70rt80rt90rr@D0@D1rrr rrr@D0@D1rr@D0@D1rr@D0@D1r
384:    rrrrr@D0@D1rrr
385:/---3.3
386:    rrr@D0@D1rr@D0@D1rr@D0@D1r rrrrr@D0@D1rrr
387:    rrr@D0@D1rr@D0@D1rr@D0@D1r
388:/---3.4
389:    r@D0@D1rr@D0@D1rrrr@D0@D1r
390:    t80rt70@D0@D1rt60rt50rt40r@D0@D1rt30r@D0@D1r
391:    t20r@D0rt50rrrrt30@D1rt40r
392:/---3.5
393:    @D0rt50rrrrrt40@D1rr t38@D0rrrrrr@D1rr
394:    t35@D0rrrrrr@D1rr t32@D0rrrrrr@D1rr
395:/---4.1
396:    t29@D0rrrrrrrr t26rrrrrrrr t23rrrrrt20rt30rrt20rr
397:    @D1rrt30@D0rrt25@D1rrrr
398:/---4.2
399:    t20rt25@D0@D1rr@D0r@D1rrrr rrrt20@D0r@D1rrrr
400:    t25@D0@D1rr@D0rr@D1rr@D0rr t20@D1rt25rrt20@D0rt25@D1rrrr
401:/---4.3
402:    t20@D0@D1rt25rt30rt35r@D0@D1rrt25rt20r
403:    @D0@D1rt30rt40rrt35@D0@D1rrt30rt25r
404:    t20@D0@D1rt30rt40rt50rt40@D0@D1rrt30rt24r
405:/---4.4
406:    @D0@D1rrrrrrrr @D0@D1rrrr@D0@D1rr@D0rr
407:    @D1rrrr@D0@D1rr@D0rr @D1rrrr@D0@D1rr@D0rr
408:/---4.5
409:    t22@D1rrrr@D0@D1rr@D0rr t21@D1rrrrrrrr
410:    t20rrrrrt34rrt30@D0rrt20rr @D1rrrrrrrrrrrrrrrr@D0
411:
412:(p)
```

リスト8　別れの曲用カウンタ表示

```
1:00001D9A 00000000   2:00001D9A 00000000   3:00001D9A 00000000   4:00001D9A 00000000
5:00001D9A 00000000   6:00001D9A 00000000
```

リスト9　ルガール総統の戦争


```
 1:/ YAMA_Ruga.ZMS
 2:.comment 宇宙戦艦ヤマト・完結編 ～ルガール総統の戦争～ for SC5
5 by Yasaka
 3:
 4:(i)
 5:(m1,2500)(aMidi04,1)          /宇宙戦艦ヤマト・完結編
 6:(m2,2500)(aMidi05,2)          /
 7:(m3,2500)(aMidi02,3)          /『ルガール総統の戦争』
 8:(m5,2500)(aMidi03,5)          /
 9:(m6,2500)(aMidi06,6)          /    作 曲  宮川  泰
10:(m7,2500)(aMidi07,7)          /
11:(m8,2500)(aMidi08,8)          /        参考CD
12:(m9,2500)(aMidi09,9)          /宇宙戦艦ヤマト Best Collection
13:(m10,2500)(aMidi10,10)        /
14:(m11,2500)(aMidi10,11)        /MIDI MODULE Roland SC-55
15:(m12,2500)(aMidi11,12)  /
16:(m15,2500)(aMidi01,15)  /
17:(m16,2500)(aMidi12,16)  /
18:
19:.roland_exclusive 16,$42 ={$40,$00,$7F,$00}
20:.sc55_reverb $10={$04,$03,$03,100,74,$00,50}
21:.sc55_chorus $10={$03,$00,$40,$08,$40,$03,$10,$35}
22:.sc55_v_reserve $10={0,2,3,4,2,2,1,2,2,2,3,1,0,0,0,0} /SC-55
23:/.sc55_v_reserve $10={1,2,3,4,3,3,1,2,2,3,3,1,0,0,0,0} /55mKII
24:.sc55_Print $10="FINAL-YAMATO"
25:
26:(t1,2,3,5,6,7,8,9) r4 @I$41,$10,$42 @G12
27:(t10,11,12,15,16) r4 @I$41,$10,$42 @G12
28:
29:/B0
30:(t1) r8 @45  @v100 @u123 @p18  r4 @E70,50 r4 t148
31:(t2) @41  r8 @v90  @u107 @p23  r4 @E65,23 r4
32:(t3) r8 @48              @p64  r4 @E55,40 r4
33:(t5) r8 @62  @v107 @u110 @p100 r4 @E50,45 @C7 r4
34:(t6) @58  r8 @v101 @u100 @p44  r4 @E60,60 @C7 r4
35:(t7) @35     @v97  @u117 @p64     @E30,0
36:      r8 x$40,$17,$32,$13,$0E,$30,$54,$0E r2
37:(t8) @74  r8 @v91  @u117 @p25  r4 @E78,61 r4
38:(t9) r8 @62  @v90  @u105 @p95  r4 @E40,45 r4
39:(t10) r16 @v110 @u127 @p64 @E127,0 x$40,$10,$15,$01 @33r16
40:/          Volume         Panpot         Rev-depth
41:        r32 @Y26,48,90  @Y28,48,97  @Y29,48,53  /H_T
42:        r32 @Y26,45,90  @Y28,45,51  @Y29,45,53  /M_T
43:        r32 @Y26,41,90  @Y28,41,51  @Y29,41,53  /L-T
44:        r32 @Y26,38,69  @Y28,38,64  @Y29,38,16  /SD
45:        r32 @Y26,36,66  @Y28,36,64  @Y29,36,16  /BD
46:        r32 @Y26,42,82  @Y28,42,68  @Y29,42,37  /C_Hat1
47:        r32 @Y26,44,82  @Y28,44,68  @Y29,44,37  /C_Hat2
48:        r32 @Y26,46,82  @Y28,46,60  @Y29,46,37  r4 /O_Hat
49:(t11) r8 r2o2l16
50:(t12) r8 x$40,$1A,$15,$02 r16 @49 @v110 @u127 r16 @p64 @E127,0
51:        r16 @Y26,57,82  @Y28,57,93  @Y29,57,80  /Con_Cym2 (o3a)
52:        r16 @Y26,38,100 @Y28,38,64  @Y29,38,64  /Con_SD(o2d)
53:        r16 @Y26,40,96  @Y28,40,64  @Y29,40,64  r8@Y24,40,63
54:        r16 @Y26,54,40  @Y28,54,100 @Y29,54,62  /Tamb (o3f+)
55:(t15) r8 @58 @v107 @u127 @p15 @M2@S4@H12 r4 @C7 @E85,75 r4
56:(t16) r8 @62 @v100 @u110 @p100 r2
57:
58:/B1
59:(t1,2,3,5,6,7,8,9) |:
60:(t10,11,12,15,16) |:
61:
62:/B2 [INT]-1
63:(t1) o6l16@Y1,99,59q6ab-ag+gg+gf+ff+fee-ee-d c+c>bb-q5aa-gg-f
64:      ee-dd-c>q4^2bb- ^3aa-^2gg-fee-dd-c>bb-q3aa-gU+8g-U-8
65:(t2) o5l16@Y1,99,59q6ab-ag+gg+gf+ff+fee-ee-d c+c>bb-q5aa-gg-f
66:      ee-dd-c>q4^2bb- ^3aa-^2gg-fee-dd-c>bb-q3aa-gU+8g-U-8
67:(t3) o2l16q3@u110@v100a
68:     {@v80@u50a@u55a@u50aZ55,50aaaaaaaa^5aaaaaaa}*180
69:     {Z55,60aaaaaaaaaZ65,70aaaaaaaaaaa}1
```

▶この時期にMPU特集はなんだかとても意味深。重大発表があるのか？
伴　武士(24)千葉県

```
70:     {Z75,80aaaaaZ85,90aaaaaZ95,100aaaaaa‾aaar}1 @u127@E34,a4
71:(t5,6,8,9,11,15,16) r1 r1 r1 r4
72:(t7,10,12) r1 r1 r1
73:
74:/B3 [INT]-2
75:(t1) _7q8@Y1,99,64@49M2@S2@M380_10a4& |:15_2a32&:|_a32r4‾42r4
76:      @M0o5q5|:{b-b-b-}4:|b-8r8{b-b-b-}4 b-8r8r2
77:(t2) _7q8@Y1,99,64@58@B8191@u127@v100@p90o0a4&
78:      a2&a8.ra4& a2&a8r8a4 a8r8@B0r2
79:(t3) @u120@v110o3|:3d8>a8:|a4 |:a4{aaa}4:| a4<@u110dd>a8<dd>a8
80:(t5) o4‾7'b-8.b<b-'q8_15|:7o5'c+24d<c+'&>'b-24b<b-':|r*44
81:      @u125@v111o4@q3|:6'b-12<b-':|'b-8<b-'r8|:3'b-12<b-':|
82:      'b-8<b-'r8r2
83:(t6) o3‾7'b8.b-'q7|:7o4'd24c+'&>'b24b-':|r*44 @v101o3@q2
84:      |:6'b-12<b-':|'b-8<b-'r8|:3'b-12<b-':| 'b-8<b-'r8r2
85:(t7) o118a2...r16 a2..r a4arr2
86:(t8) r1 r2.q5o6|:3'f12b-':| 'f8b-'r8r2
87:(t9) @Y1,100,114@v100@u100o5'b-8.b'&@B1366'b-24b'&|:6@B0
88:      'b-24b'_20@B1366'b-24b'&‾20:|_20@B1366'b-24b'
89:      @v0@B0r*44 r1 r2.
90:(t10) o2c4r2. c4r2. r1
91:(t11,15,16) r2. r1 r1
92:(t12) o3a4r2. a4r2. r1[K.SIGN +f,]
93:
94:/B4 [INT]-3
95:(t1) o3@E90,60@q2 |:r2@v103a_8b-ab-ar8. |r1:| r2@41@E55,80r2
96:(t2) @49@p104@u100@v85o2q8|:r2ab-ab-ar8. r1:| @45
97:(t3) r1 r1 r1 @E58,r1@u110@v65
98:(t5,6,8,11,16) r1 r1 r1 r1
99:(t7) 116_20|:r2ar8.ar8. r1:|‾20
100:(t9) r1 @Y1,100,64r1 @74@v90@u105@p30@E78,45r1 r1
101:(t10) r1 r1 r1 r2o3116@u120cc@v75c8@v107>aa@v80@B2500fr@B0
102:(t12) o2124@v60d@v22|:11ed:|e |:18de:| |:‾2ded ‾ede:| |:3‾3ded
103:       |‾ede:| ‾5ede ‾ded ‾ede
104:(t15) o3@L2@q1f+gg+a*42<eff+g*18r8r2 >f+gg+a*18<eff+g*54>r16|:
105:       fgg+a*18<eff+g*18>:| f+*3g*3g+*3a*39<e*3f*3f+*3g*19r*116
106:       >f+gg+a*18<e*3f*4f+*3g*38& |:15_2g32&:|_g32
107:
108:/B5 [INT]-4
109:(t1) @u110@v70@p45@Y1,99,58o3q5|:4defg agfe defgagfe:|
110:(t2) @E55,90@u115@v82@p30q4o4@Y1,99,59|:4defg agfe defgagfe:|
111:(t3) o3|:4Ud4r8>U-10aa<Ud4r8>U-10aa:|
112:(t5) r1 r1 r1 @E118,53r2@Y1,99,60@Y1,100,90r*95
113:(t6) r1 r1 r1 @E95,80r1
114:(t7) @v103o2q8dr4r>ar<drr4>ar <|:3drr4>ar<drr4>ar:|
115:(t8,9,16) r1 r1 r1 r1
116:(t10) @v100 @u120o214|:4cdcd:|
117:(t11) [K.SIGN +f,+g,-b]
118:       Z100,60,65,70,,60,85,60,70,,60,90,70,60,,70
119:       |:4fffg ffff gfff ff|gf:|fb
120:(t12) @v100o3116|:Z90,120a*0&f8@u90ff||:3@u120f8@u90ff:|:|
121:       |:@u120f8@u90ff:|@u90ffff |:7@u120f8@u90ff:|@u90ffff
122:(t15) r1 @v97r1 o3@q1r16f+gg+a*42<e*3f*3f+*3g*123
123:       r*12>f+gg+a*42<eff+q6g*42>q8f+gg+a*18<e*3f*3f+*3g*51
124:
125:/B6 [A]-1
126:(t1) @v63|:defgagfe:| _8|:6'd<d''e<e''f<f''g<g''a<a''g<g''f<f'
'e<e':|
127:(t2) @v75o4|:defgagfe:| ‾3q4|:4o4b-<cdefedc:| |:o4ab<cde-dc>b:
|
128:(t3) o3d4r4r2 r1 r1 r1
129:(t5) @u112@v101@p105o4116q6a8r<q5U-11dUq6d8rq5U-12c+q6Uc+8rq5U
-10cq6Uc8r>q5U-10b- q6Ub-8r<q5U-10fq8 Uf&_80f&|:9‾20f&:|f&f1 r1
130:(t6) @p20r1 r2.@u115@v105o4116q6d8rq5a-
131:      q8a-2&|:7_3a-&:|_3a- r2@62@u110@v87@p115r2
132:(t7) o2dr4r>ar<drr4>fr |:b-rr4frb-rr4|fr:|<dr grr4drgrr4>b-r
133:(t8) r1 r1 @u94@v80r1 o6116q6d8rgg8rf+f+8rff8re-
134:(t9) r1 r1 @u86@v55r1 o5116q6d8rgg8rf+f+8rff8re-
135:(t10) |:4cdcd:|
136:(t11) fffg ffff gfff ffgf b8fg ffff gfff ffgf
137:       Z100,60,65,70,,60,85,60,70,,60,90,70,60,,70
138:       |:fffg ffff gfff ffgf:|
139:(t12) @u120a*0&|:5@u120f8@u90ff:| @u100a*0&|:7@u120f8@u90ff:|
140:       @u120a*0&|:3@u120f8@u90ff:|@u90ffff

141:(t15) r1 r1 r1 r1
142:(t16) @p100r1 @E50,34r1
143:       @u120@v90q8r4o4116f32g32@B0,-2049,30a-8.&_12a-&
144:
145:/B7 [A]-2-1
146:(t1) |:4'd<d''e<e''f<f''g<g''a<a''g<g''f<f''e<e':| @E117,
147:(t2) |:4o4ab<cde-dc>b:|
148:(t3,15) r1 r1
149:(t5) r1 @v90r4>q6@u101a32<q8U+9c+.r8>U-4b-&_60|:8‾15b-32&:|b-8

150:(t6) r2.o3q6b-8r<q5c+ q8c+1 @E50,80 @58 @p25
151:(t7) |:<e-rr4>b-r<e-rr4>b-r:|
152:(t8,9) e-8rb-q8b-2.& |:15b-&_3:|_b-
153:(t10) ddcd cdcd
154:(t11) b4b4 gfff ffgf ffff ffff gfff fffb
155:(t12) |:Z88,120a*0&f8@u90ff:||:6@u120f8@u90ff:|
156:(t16) |:4‾12a-&:|a-8. r1 @B0r1 @45@u100@v50@E80,70@p20r1
157:
158:/B8 [A]-2-2
159:(t1) o5q8@v53aa-q6arq8f+fq6f+rq8e-dq6e-rr4
160:      q8c>b<q6cr>q8aa-q6arq8f+fq6f+rr8@E55,r8
161:(t2) @E55,o5q5@v110aa-arf+ff+re-de-rr4 c>b<cr>aa-arf+ff+rr4
162:(t3) ‾10|:o3fffrfffr>aaarr4:|_10
163:(t5) @E55,53@u107@v74|:o4_7'ea-b-'r8.‾7r2>'b->b-'r'a>a'r:|
164:(t6) @u114@v80|:o4_7'ea-b-'r8.‾7r2>'b->b-'r'a>a'|r:|r*11
165:(t7) o1|:r2.b-rar:|
166:(t8,9,11,15) r1 r1
167:(t10) @u86r2.a8f8 r2.a8f8
168:(t12) |:@u100a8r8r2@u60f8f8 |@Y28,57,33:|
169:(t16) o5q8116f+ff+re-de-rc>b<crr4 >ag+arf+ff+re-de-rr8r8
170:
171:/B9 [B]-1
172:(t1) @u110@v62@p45o3q5|:defgagfe defgagfe:|
173:(t2) @E55,@u115@v75@q30q4o4|:defgagfe defgagfe:|
174:(t3) o3|:d4r8>U-10aa<Ud4r8>|U-12aaU:|o2U-12ffU
175:(t5) @E110,53@u112@v87o4q6a8r<q5U-10dq6Ud8rq5U-11c+q6Uc+8rq5U-
9cq6Uc8r>q5U-10b- q6Ub-8rU-10aq8Ua&_70a32&|:10‾15a&:|‾6a32
176:(t6) @E45,80@u94@v98o3q6a8r<U-6dUd8rU-7c+Uc+8rU-6cUc8r>U-8b-
177:      Ub-8rU-5aq8Ua&a32&|:10_3a&:|a32
178:(t7) o2|:dr8.r8>ar<drr4>|ar:|o1fr
179:(t8) @u92@v78q8o6r1 |:defgagfe:|
180:(t9,15,16) r1 r1
181:(t10) @u120ddcd cdcd
182:(t11) Z100,60,65,70,,60,85,60,70,,60,90,70,60,,70
183:       |:fffg ffff gfff ff|gf:|fb
184:(t12) @u120a*0&|:8@u120f8@u90ff:| @Y28,57,93
185:
186:/B10 [B]-1-2
187:(t1) _8|:4'd<d''e<e''f<f''g<g''a<a''g<g''f<f''e<e':| @E117,
188:(t2) q4‾3|:o4b-<cdefedc>b-<cdefedc:|
189:(t3) o2|:b-4r8U-12ffUb-4r8U-10ffU:|
190:(t5) @E40,53@u105@v88o3q6b-8r<q5U-8fq6Uf8rq5U-9eq6Ue8rq5U-11
191:      e-q6Ue-8rq5U-8d Ud8r>U-10b-q8Ub-4&|:7_8b-&:|_b-*13
192:(t6) @E95,80@u120@v121o3q6b-8r<U-6fUf8rU-6eq6Ue8rU-6e-Ue-8r
193:      U-7d Ud8r>U-5b-q8Ub-4&|:7_8b-&:|_b-*13 @E50,80
194:(t7) |:b-rr4frb-rr4fr:|
195:(t8) r1 |:o5b-<cdefedc:|
196:(t9,15,16) r1 r1
197:(t10) ddcd cdcd
198:(t11) b4 b4 gfff ffgf ffff ffff gfff ffbf
199:(t12) |:3Z78,120a*0&f8@u90ff:| |:5@u120f8@u90ff:|
200:
201:/B11 [B]-2-1
202:(t1) o5q8@v53aa-arf+ff+re-de-rr4 c>b<cr>aa-arf+ff+rr4
203:(t2) @E55,o5q5@v110aa-arf+ff+re-de-rr4 c>b<cr>aa-arf+ff+rr4
204:(t3) ‾10|:o3fffrfffr>aaarr4:|_10
205:(t5) @E55,53@u107@v74|:o4_7'ea-b-'r8.r2>‾7'b->b-'r'a>a'r:|
206:(t6) @58@p25@u114@v80|:o4_7'ea-b-'r8.r2>‾7'b->b-'r'a>a'r:|
207:(t7) o1|:r2.b-r16ar:|
208:(t8,9,11,15) r1 r1
209:(t10) @u86r2.a8f8 r2.a8f8
210:(t12) |:@u100a8r8r2@u55f8f8:|
211:(t16) o5q8116f+ff+re-de-rc>b<crr4
212:       >ag+arf+ff+re-de-rr8@B-64@E75,80@p110r8
213:
214:/B12 [B]-2-2
215:(t1) o3@E75,q4@u105@v60aab-b-bb<ccc+c+dd‾6e-e-‾ee
216:      ‾ff‾f+f+‾ggg+g+q6ar8.r@u90@v78q8@Y1,99,64r8.
217:(t2) o3@E60,q4@u106@v100'a<a''a<a''b-<b-''b-<b-''b<b''b<b'<
218:      'c<c''c<c''c+<c+''c+<c+''d<d''d<d'_3'e-<e-''e-<e-'_'e<e'
219:      'e<e'_'f<f''f<f' _'f+<f+''f+<f+'_'g<g''g<g''g+<g+'‾g+<g+'
220:      q6'a<a'r4@49@u100@v105q8@Y1,99,64r8.
221:(t3,10,11,15) r1 r1
222:(t5) @v77r1 r2.o4@u87g64g+64@u110a.r8
223:(t6) @v84r1 r2.o4@u84g64g+64@u114a.r8
224:(t7) r1 r2.a8r8
225:(t8) r1 r2@u100@v87r4o5'g64<g''g+64<g+''a.<a'r8
226:(t9) r1 @14@v92@u88@p74q8@E60,35
227:      r2.o5g64g+64a.r@49@p105@u93@v76r@E80,60
228:(t12) r1 r2.@u100a8r8
229:(t16) o3q4@u105@v78116aab-b-bb<ccc+c+dd‾6e-e-‾ee
230:       ‾ff‾f+f+‾ggg+g+q6ar8.r4@B0
231:
232:/B13 [C]-1
233:(t1) @p35@q1o4a4.<U-5d8d4.c+8 c+4.c8c4.>Ub8 b-4.a8d2&
234:      d4e8f8g8f8q4@u106e8d8
235:(t2) @p30@q1o5a4.<U-5d8d4.c+8 c+4.c8c4.>Ub8 b-4.a8d2&
236:      d4e8f8g8f8q4@u117e8d8
237:(t3) @v68o2|:8@u108f4@u117a4@u108f4@u117a4:|
238:(t5) r1 @E100,53r1@u107@v100@p105o4q6a8r<q5U-10dq6Ud8rq5
239:      U-11c+q6Uc+8rq5U-9cq6Uc8r>q5U-10b-
240:      Uq6b-8rq5U-10aq7Ua2.@Y1,99,64
241:(t6) r1 r1 @E80,50@u115@v90r1 r1
242:(t7) _5o2|:4q6d8q4e4q8f.r32q6f8q4e4q6d8:|
243:(t8,16) r1 r1 r1 r1
244:(t9) o318a4.<dd4.c+ c+4.cc4.>b b-4.ad2& d4efgfq4@45@u114ed@q1
245:(t10,11) |:8r1:|
246:(t12) |:3@u73a*0&|:7@u105f8@u77ff:|@u77ffff:|
247:(t15) @E40,q8r8@p110@u127@v65o1|:a8r4._15|:4a8&‾4:|
248:       |:3‾a8&:|‾a8&|:7_4a16&:|_4a16:|
249:
250:/B14 [C]-2
251:(t1) @q1@u97 f4.>@u93 q7b-8@q2@u85b-2&
252:      b-4@u90 <c8d8e8d8q4@u110c8>b-8 <@q1@u100c+4.>q7@u94
253:      a8q8@u83a2& a2&|:7_3a16&:|_3a16
254:(t2) @q2@u106f4.>@u103q7b-8@q2@u95b-2&
255:      b-4@u100<c8d8e8d8q4@u120c8>b-8 <@q1@u110c+4.>q7@u103
256:      a8q8@u95a2& a2&|:7_3a16&:|_3a16
257:(t5) @p20o4@u90_30fr8.‾20fr8.‾10fr8.q4f8b-8 {a-a-g}4
258:      q8f2&|:3‾5f32&:|‾5f32_20r8 @p105q5@u117o4|:4r4gggr:|
259:(t6) o4_30fr8.‾20fr8.‾10fr8.q4f8b-8 {a-a-g}4q8f2&|:3‾5f32&:|
260:      ‾5f32_20r8 @u118‾12q5o4|:4r4eeer:|
261:(t7) >|:q6b-8<q4c4q8d.r32q6d8q4c4>q6b-8:| q6a8q4b-4<q8c.r32
262:      q6c8>q4b-4q6a8 a8<q4g4q8f.r32q6f8q4e4q6d8
263:(t8) r1 r1@u96@v87 @p85o5q6a8r<ee8rdd8rc+c+8r>b- b-8ra<g2.
264:(t9) @u98f4.>q6@u105@49b-@q2@u92b-2& b-4@u93<cdedq4 @45@u116
265:      c>b- <@q1@u100c+4.>q7@u105aq8@u94a2& a2&|:7_3a16&:|_3a16
266:(t12) |:@u73a*0&|:4@u105f8@u77ff:|:|
267:(t15) @v65o1b-8r4._15|:4b-8&‾4:| |:3‾b-8&:|‾b-8&|:7_4b-16&:|
268:       _4b-16 @v40o1b-8r4._10|:4b-8&‾4:|
269:       |:3‾b-8&:|‾b-8&|:7_4b-16&:|_4b-16@p15@E85,
270:(t16) r1 r1 r1 r1
271:
272:/B15 [D]-1
273:(t1) r1 @u100@v70q4o5r2r8g8<c8>b8
274:(t2) r2@45r2 @u110@v85q6o5r2r8g8<c8>b8
275:(t3) @v110o2Z50,55{b-b-b-b-b- b-b-b-b-b- b-b-b-b- b-b-b-b-}1
276:      {Z58,63b-b-b-b-b- Z66,71b-b-b-b-}2
277:      {Z74,79b-b-b-b-b- Z82,87,90,95b-b-b-b-}2
278:(t5) @v88o2q4r8b-8<f8e8q8e-2& e-1
279:(t6) @v115o3q4r2r8g8<c8>b8 _10q8b-1
280:(t7) _10@q2>r8b-8<f8e8q8e-2& e-1
281:(t8) o3q4@62@p25@v100r2r8g8<c8>b8 q8b-1
282:(t9) r2@62@u110@v100@p45r2 q4o4r'c<c''f<f''e<e'q8'e-2<e-'
283:(t10) @u120c16r8.r2. r1
284:(t11,15) r1 r1
285:(t12) @u100ar2... r1
```

▶定期試験直前に「GAME OF THE YEAR」はあまりにも“毒”すぎる。ああ，試験が，単位が……。
小橋　玄央(21)茨城県

```
286:(t16) @62@E50,10r1 @u110@v105q4o4r2r8'g8<g'<'c8<c'>'b8<b'
287:
288:/B16 [D]-2
289:(t1) @u106o4q6|:b-6b-6|b-6:|q8b-12r24@p18@u123@v90@E70,r24
290:(t2) o4q6|:6'b-6<e':|
291:(t3,11,15,16) r1
292:(t5) @E60,53o4q6@u110@v104|:6'b-6<e':| r4q8
293:(t6) o3q6@u116@v95|:6'e6b-':| r4 q8
294:(t7) ~16>q6{b-b-b-b-b-b-}1
295:(t8) @73@u100@v80o6q6|:6'e6b-':|
296:(t9) o5q6|:6'e6b-':|
297:(t10) |:6'c6d':|
298:(t12) @u70{aaaaaa}1
299:
300:/B17 [D]-3
301:(t1) q8@49M2@S2@M380o3a4& |:15_2a32&:|_a32r4~42@E45,r4
302:(t2) @58@u115@v100@B8191@p90q8o0a4& a2&a8.ra4&
303:(t3) r4 q4@u110@v90o3d8>a8<d8>a8<d8>a8a8r8
304:(t5) @v109o4'b-8.b<b-'q8_15|:7o5'c+24d<c+'&>'b-24b<b-':|r*44
305:(t6) @v97o4b-8.&@B1998b-24|:7@B0b-24&_20@B1998b-24~20:|@B0r8.
306:(t7) @v97@u117q8a4& a2&a8.ra4&
307:(t8,11,15,16) r4 r1
308:(t9) r4@Y1,100,114@62@p105@v100@u100o5'b-.b'&@B1366'b-24b'&
309:      |:6@B0'b-24b'_20@B1366'b-24b'&~20:|_20@B1366'b-24b'@v0
310:      @B0r*44@y1,100,64
311:(t10) c16r8. r2.c16r8.
312:(t12) @v107@u127ar8. r2.ar8.
313:
314:/B18 [D]-4
315:(t1) @M0o5q5|:{b-b-b-}4:|b-8r8{b-b-b-}4 b-8r4.@E25,@Y1,99,58r4
316:(t2) a2&a8r8a4 a8r4.@E45,78@B0@45@Y1,99,58r4
317:(t3) a8r8{aaa}4a12r6{aaa}4 a12r6<@u100dd>a8<dd>a8
318:(t5) @E40,53@q3@u125@v109o4|:6'b-12<b-':|'b-8<b-'r8
319:      |:3'b-12<b-':| 'b-8<b-'r8@E60,53r2
320:(t6) @q3@v105|:6'b-12<b-':|'b-8<b-'r8|:3'b-12<b-':|
321:      'b-8<b-'r8@E35,50r2
322:(t7) a2&a8r8a4 a8r8r2 132
323:(t8) @74@u105@v103r2.q5o6|:3'b-12f':| 'b-8f'r8@p95r2
324:(t9) @q3@u110@v88o3|:6'b-12<b-':|'b-<b-'r|:3'b-12<b-':|
325:      'b-<b-'rr2@62
326:(t10,11,12,15,16) r1 r2.
327:
328:/B19 [E]-1
329:(t1) o6@u100@v90@p20@q3{defedefdc+dedc+ded}*198
330:      {defedefdcde-dcde-c}*189 @u104@q4
331:      {>b-<cdc>b-<cd>b-ab-<c>b-ab-<c>b-}*189
332:      @u108@q5b-<cdc>@q6b-<cdc>b-r8.@Y1,99;64r4
333:(t2) o5@u100@v85@p20@q3{defedefdc+dedc+ded}*198
334:      {defedefdcde-dcde-c}*189 @u104@q4
335:      {>b-<cdc>b-<cd>b-ab-<c>b-ab-<c>b-}*189
336:      @u108@q5b-<cdc>@q6b-<cdc>b-r8.@Y1,99,64r4
337:(t3,8,9,12,16) r1 r1 r1 r1
338:(t5) @v80o4q6d4.@q4U-23dU+23dq6e4.@q4U-28eU+28e
339:      q6f4.@q4U-28fU+28fq6f+4.@q4U-23f+U+23f+
340:      q6g4.@q4U-23gU+23gq6a4.@q4U-23aU+23a q4b-8r<c8rd8e8r4.
341:(t6) @H,36@A110,,,,,100,90,80@p100@u110@v110o3
342:      q7d4.@q4U-23dU+23dq7e4.@q4U-28eU+28e
343:      q7f4.@q4U-28fU+28fq7f+4.@q4U-23f+U+23f+
344:      q7g4.@q4U-23gU+23gq7a4.@q4U-23aU+23a
345:      q4b-8r<c8rd8e8r4@A@p30r8
346:(t7) @q3<d8..U-20dUa4e8..U-20eUa4 f8..U-20fUa4f+8..U-20f+Ud4
347:      g8..U-20gU<d4>a8..U-20aU<d4> g8..U-20gU<d4>>g8r4.
348:(t10) |:4cd|cd:|r2
349:(t11) 116|:3|:2@u80b8f@u60f@u65b8@u30b@u85f:|:|
350:      @u100b8f@u85f@u115b8.@u127bfr4..
351:(t15) r1 r1 r1 @A80,90,100,110,,,,r1
352:
353:/B20 [E]-2
354:(t1) @u85@v56q8o4|:'a<e'rr2'a<e'r'b-4<f':|
355:(t2) r1 @62@p110@u110@v100@E40,q8r1
356:(t3) |:@u100ar8.r4.arZ55,60{b-b-b-b- b-b-b-b-}4 :|
357:(t5,6) @u100@v100q8o3|:'a<ea'rr2'a<ea'r'b-4<fb-':|
358:(t7) 116~10arr2arb-4 arr2arb-4
359:(t8) @u96@v75q8o6|:'ea'rr2'ea'r'f4b-':|
360:(t9) @u110@v78@p20o4116r4ra2r8. r4ra2r8.q6
361:(t10) c16r8.r2. c16r8.r2.
362:(t11,16) r1 r1
363:(t12) @u100|:ar2...:|
364:(t15) o4116@u120@v115|:r4ra2r8.:|
365:
366:/B21 [E]-3
367:(t1) 'a<e'r4r'b-<f'r'a<e'r4r'b-<f'r
368:      'a<e'r8.'b-4<f''a<e'r'a<e'r8.@45@v100@u123@E70,@p18r8
369:(t2) q6o5|:ar4rb-r:| ar8.q8b-4q6arar8.@41@v90@u107@p23@E65,r8
370:(t3) @u100ar8.r8arar4^16ar ar8.Z60,65{aaaaaaaa}4 Z@u100ararr4
371:(t5) U+10|:'a<ea'r4r'b-<fb-'r:| 'a<ea'r8.'b-4<fb-''a<ea'r
372:      'a<ea'r8.@v107@u110@p100@E50,45@Y1,100,64r8
373:(t6) |:'a<ea'r4r'b-<fb-'r:| 'a<ea'r8.'b-4<fb-''a<ea'r
374:      'a<ea'r8.@v101@u100@p44@E60,60r8
375:(t7) |:ar4rb-r:| ar8.b-4arar8.@v97@u117@E30,0r8
376:(t8) ~15|:'ea'r4r'fb-'r:| 'ea'r8.'f4b-''ea'r'ea'r8.
377:      @v91@u117@p25@E78,61r8
378:(t9) |:r8aaar8.:| q4r8a8r8a8q6arar8.@62@v90@u105@p95@E40,45r8
379:(t10) c16r8.rc16r8.r c16r8.r2.@v110@u127
380:(t11) r1 @u110r4br8@u50fbfbfr4
381:(t12) ar8.r4ar8.r4 ar2...@v107@u127
382:(t15) q6r8aaar4raaar4r q4a8r8a8q6arar8.@A
383:      @v107@u127@M2@S4@H12@M0@p15@E85,75r8
384:(t16) r1 @E90,10r1
385:
386:/B22 B2へ戻る
387:(t1,2,3,5,6,7,8,9) :|
388:(t10,11,12,15,16) :|
389:
390:/B23 SC-33では、t5のI127 @97をI0 @62に変更して下さい
391:(t1) @49@u100@v70o516q6|:3de-eff+|g:|q8g& g1
392:(t2) @59@p80@u100@v8016q6|:3o4dc+c>bb-|a:|o3q8a& a1
393:(t3) @u10016q4|:3o3dc+c>bb-|a:|@v115@u50a116
394:      Z55,60aaaaZ65,70aaaaZ75,80,85,90,95,100,105,110aaaaaaaa
395:(t5) I127@97@u110@v125o516q6|:3'df''e-f+''eg''fg+''f+a'|'gb-':
|
396:      q8@u120'gb-'& 'g4b-'&|:11~10'g16b-'&:|'g16b-'
397:(t6) @u110@v10516q6|:3o2'd<d''c+<c+''c<c'>'b<b''b-<b-'|'a<a':|
398:      q8o1'a<a'& 'a1<a'
399:(t7) o2~1016q6|:3o2dc+c>bb-|a:|o1q8a& a1
400:(t8) @u102@v95o616q6|:3de-eff+|g:|q8g& g1
401:(t9) @62@u110@v9516q6|:3o4dc+c>bb-|a:|o3q8a& a1
402:(t10) 16o2@u100|:3cccccc:| r1
403:(t11) r1 r1 r1 Z40,,55,,60,65,70,75,80,85,90,95116|:3bbbb:|
404:      @u100bbbb
405:(t12) 16q2@u85o3|:3aaaaaa:| r2.@Y28,57,64r8.@u90a16
406:(t15) r1 r1 r1 r1
407:(t16) @B-65@u110@v95o416q6|:3'df''e-f+''eg''fg+''f+a'|'gb-':|
408:      q8'gb-'& 'g4b-'&|:11~10'g16b-'&:|'g16b-'
409:
410:(p)
```

リスト10 宇宙戦艦ヤマト用カウンタ表示

```
 1:000051A8 00000000   2:000051A8 00000000   3:000051A8 00000000   5:000051A8 00000000
 6:000051A8 00000000   7:000051A8 00000000   8:000051A8 00000000   9:000051A8 00000000
10:000051A8 00000000  11:000051A8 00000000  12:000051A8 00000000  15:000051A8 00000000
16:000051A8 00000000
```

リスト11 おまけ

```
 1:/ YAMA_KEI.ZMS
 2:.comment ヤマト戦闘時の艦内警報音? (SC-55対応)
 3:
 4:(o125)
 5:(m1,1000)(aMidi01,1)/このデータは戦闘時の曲とミックスして
 6:(m2,1000)(aMidi02,2)/聴いて下さい(これだけで聴かないでね)
 7:(m3,1000)(aMidi03,3)
 8:(m4,1000)(aMidi04,4)
 9:(m5,1000)(aMidi05,5)
10:(m6,1000)(aMidi06,6)
11:
12:.roland_exclusive 16,$42 ={$40,$00,$7F,$00}
13:.sc55_reverb $10={$04,$03,$07,100,74,$00,50}
14:.sc55_chorus $10={$03,$00,$40,$08,$40,$03,$10,00}
15:.sc55_v_reserve $10={6,6,4,4,2,2,0,0,0,0,0,0,0,0,0,0}
16:
17:(t1,2,3,4,5,6) @I$41,$10,$42 Y11,64 @p64 @u127 r4
18:(t1,2) @G7  @E0,20 I8 @20  @v100 r4 @Y1,32,80 @Y1,33,85 r4
19:(t3)   @G6  @E64,30   @61  @v95  r4 @Y1,32,80  r4
20:(t4)   @G12 @E0,40    @80  @v110 r4 @Y1,33,100 r4
21:(t5,6) @G7  @E0,20    @110 @v120 r4 @Y1,99,68 @Y1,32,74
22:       @Y1,33,85 r*46
23:(t1) [DO] @B0,8191,16o4'ce<ce'*190&'ce<ce'*4 r2r*94 [LOOP]
24:(t2) [DO] r2r*94@B0,8191,16o4'ce<ce'*190&'ce<ce'*4 [LOOP]
25:(t3) [DO] @B0,8191,16o4'c2.e'r4 [LOOP]
26:(t4) [DO] o6|:16u-50'e-16e'u'b16<c':|[LOOP]
27:(t5) [DO] @B0,8191,16o3'ce'*190&'ce'*4r2r*94 [LOOP]
28:(t6) [DO] r2r*94@B0,8191,16o3'ce'*190&'ce'*4 [LOOP]
29:
30:(p)
```

リスト12 おまけ用カウンタ表示

```
1:00000090 00000180   2:00000090 00000180   3:00000090 000000C0   4:00000090 00000180
5:0000008E 00000180   6:0000008E 00000180
```

▶PowerPC620やSH2よりもMC68060のほうがいい〜。でも，080は出ないんだろうな。
中上　匠(18)千葉県

(善)のゲームミュージックでバビンチョ

西川善司

●ファルコムスペシャルBOX'95
CD:KICA-9023～5　7,400円(税込)
キングレコード　発売中

毎年恒例の3枚組のファルコムのお祭りCD。DISC1がCDドラマ風のザナドゥ，DISC2が英雄伝説IIIオーケストラアレンジバージョン，DISC3がイースIV。

今回特におすすめしたいのがDISC2だ。オーケストラアレンジという企画自体はありがちなのだが完成度がそんじょそこらのものとは違う。映画音楽を思わせる壮大なものから，ボストンポップス(オーケストラで軽音楽を演奏する楽団)を彷彿させるJAZZYなものまでバラエティに富んだ内容，アレンジャーの天才ぶりが至るところに垣間見られる。メロディなども大変美しく，ゲームを知らなくても十分楽しめるCDだ。私もゲームはよく知らないのだがこのCDは実に楽しく聴くことができた。交響曲ファンにはぜひ聴いてもらいたい。はたしてこれは打ち込み曲なのだろうか。「演奏：JDKエレクトリックオーケストラ」とあるから打ち込みなのだとは思うが。

おすすめ度　8 (DISC2は10)

●ポリスノーツ/コナミ矩形波倶楽部
CD:KICA-7653　2,800円(税込)
キングレコード　発売中

私が昨年プレイしたゲームのなかでもっとも印象に残っているゲームだ。完成されたストーリー，アニメや映画では絶対味わえないインタラクティブな演出の数々。CD-ROMの容量にモノをいわせた大量の音響演出(リアルな効果音/環境音，声優起用による映画さながらの台詞まわし)も，ほかのCD-ROMゲームのそれと比べると突出した完成度。

今回発売されたCDはゲーム中に使用されたBGMのオリジナルを全曲収録したもの。アドベンチャーゲームというジャンルの性質上，音楽も情景描写系の「BGM」が多い。だが，さすが難波の吟遊詩人矩形波倶楽部。どの曲も文句なしにカッコイイ。フュージョンやムード音楽系の曲は，いつもの，CDと一緒に歌いたくなってしまうような痺れるメロディアスなあのパターン。ところで演奏はどうもSC-88くさい……。

おすすめ度　10

●リッジレーザー
/namcoサンプリングマスターズ
CD:VICL-15038　1,500円(税込)
ビクターエンタテインメント　2/22発売

今回発売されたのは「リッジレーサー２」アレンジバージョン＋αのアルバム。アレンジバージョンとはリッジレーサー/リッジレーサー2のLD(ビデオ)に収録されていた5曲のこと。そして今回のCDのための4曲の書き下ろし曲が「＋α」。レーザーディスクの曲ということでCDのタイトルは「リッジ『レーザー』」。

音楽を聴く場合にはやはり聴く側にそれを受け止める器(センス?)が要求されると思う。そしてそのあたりが「好み」というものに繋がっていくのだろう。というのも，以前の私ならばこのたぐいの音楽(デステクノ?)はまるっきしダメだったのだが，最近PlayStationで「リッジレーサー」をプレイするようになってから，ちょっとだけわかるようになってきたのだ。

確かにリッジレーサーという暴力的なスピードとのスリリングな駆け引きを楽しむゲームにはこういう凄まじいまでの音の嵐の音楽がよく似合っている。むしろこれしかないという気までする。そういうわけで，ゲームをやっているうちに「曲がゲームにハマってるなぁ」と感心してこれが気持ちよく聞こえてきて，いつのまにか「イイ」と思えるようになってしまったのだ。で，このCDも「イイ」のだが万人向けではない。これだけはいえる。

おすすめ度　8

●ツインビーPARADISE2 Vol.3
CD:KICA-7655　2,800円(税込)
キングレコード　2/22発売

感動の最終回で惜しまれつつ幕を閉じた「ラジオドラマ版ツインビー」のパート2が昨年秋からオンエアされていた(文化放送系列)。もちろんその番組を収録したこのCDシリーズも発売されていて，いつのまにかもうVol.3になっている。Vol.1～3まで聴いた感じでは毎回CD1枚で話が完結する構成で，さらにCD終盤にはイメージソング&出演声優さんたちのフリートーク(サイコロトーク!?)のおまけが収録されている。

今回のドラマは國府田マリ子扮するマドカが海底文明と接触するSFちっくなお話。おまけのほうも國府田マリ子がメインだ。

おすすめ度　7

●ぽっぷるメイルパラダイス 2
CD:KICA-1156　3,000円(税込)
キングレコード　2/22発売

ツインビーに負けじとファルコムのコミカルARPGの「ぽっぷるメイル」もドラマに。こちらはラジオ放送の収録ではなくCDのための録り下ろし作品だ。ドラマは3部構成で前回同様CD1枚で完結。ドラマの合間合間に同ゲームのBGMのアレンジバージョンが収録されている(全5曲/内ボーカル1曲/カラオケ1曲)。

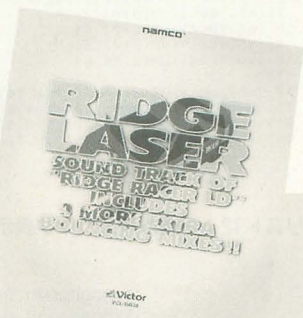
賞金首を追って雪山へ入ったメイルは遭難そのうえ雪崩にあってしまう。そして無一文で麓の温泉宿に泊まるが……という感じのゲーム本編とはあまり関係のないナンセンスドタバタコメディ。

ところで出演している声優の顔ぶれが「ツインビーPARADISE」に似ているのはやっぱりいろんな事情があってのことなのかしらん。

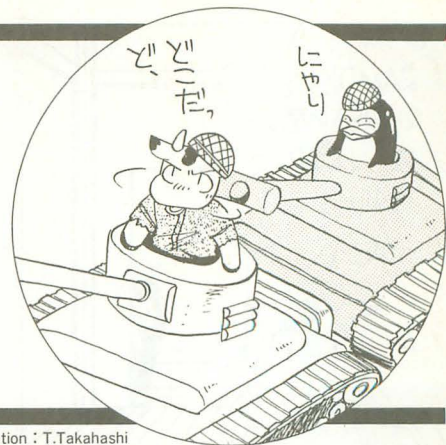
おすすめ度　7

ポリスノーツ

リッジレーザー

(で)のショートプロぱーてぃ——その66

対戦ゲームだフィールドバトル

Komura Satoshi 古村 聡

天災は忘れた頃にやってくる，ということで身の周りを確かめてしまった（で）氏ですが，皆さんも安全対策は万全にね。今月は，ゲーム２本にツールを１本を紹介します。どれも結構必要とするものが多いので，注意してください。

illustration：T.Takahashi

1995年１月17日。淡路島を震源とする大地震により，神戸の街が建物の倒壊と火災で大変な被害を受けて，とんでもない状況となっていましたね。テレビでその情報を知った私は，思わず部屋に置いてある非常用袋のカンパンと水を点検してしまいました。明日は我が身，他人ごとではありませんからね。でも，考えてみたら，私の部屋はベッドの足元と頭の横に２つディスプレイを載せたパソコンラックがあったりして……もし，ここで神戸と同じような地震が起きて，これらのものが頭に落ちてきたら命がないですよね，やっぱり。なんてったって，重さが5kg，15インチのガラス張りなわけだから……。

しかも，チルトスタンドまでつけてあって，なんか，わざわざ落ちやすいようにしてるようなものだな，これは。ちょっとぞっとしてしまいます。もっとも，家具なんかには耐震補助金具などをつけたりしますけれども，ディスプレイにはそういうものもないわけで……，まさか，ラックとディスプレイをボルトで固定するわけにもいかないし。出ませんかね，耐震構造ディスプレイ。

冗談はともかく，Oh!Xに愛読者ハガキをくださる関西圏の皆さんは大丈夫でしたか？ ショートプロへプログラムを投稿してくださる方の中にも関西圏の方が何人かいるのでちょっと心配です。無事だったらぜひ愛読者ハガキでもくださいね。

タンクでバトル，フィールドバトル

では，プログラムにまいりましょうか。今月は……あ，２カ月連続で登場の佐怒賀さんのゲームプログラムですね。どんどんバコバコ，タンクでバトル，対戦バトルフィールドゲーム，BATTLE.BASです。どうぞっ。

BATTLE.BAS for X680x0
(X-BASIC，要XSPRITE.FNC/ZMUSIC.FNC/ZMUSIC.X)

神奈川県 佐怒賀 英一

このプログラムはX-BASIC用のプログラムですが，実行するためにはスプライトを簡単に制御するためのXSPRITE.FNCと，Z-MUSIC用のBASIC外部関数，MUSICZ.FNCがX-BASICに組み込まれている必要があります。

XSPRITE.FNCをX-BASICに組み込むには，X-BASIC本体の置かれているディレクトリ(デフォルトだとBASIC2という名前になっているはずです)にある，X-BASICの外部関数の組み込みを設定するファイルにXSPRITE.FNCが組み込まれることを書いておく必要があります。具体的には，そのディレクトリにXSPRITE.FNCをコピーしておいてから，ED.Xなどのエディタで，BASIC.CNFというファイルに，

```
FUNC = XSPRITE
```

と書けばOKです。

また，MUSICZ.FNCの場合には，外部関数を組み込んでから，Z-MUSICを使える状態にしなければなりません(Z-MUSICを制御するための外部関数なんだから，当然といえば，当然ですね)。MUSIC.FNCはXSPRITE.Xと同様にBASIC.CNFへ，

```
FUNC = MUSICZ
```

の１行を追加してからコマンドラインなどから，

```
A>ZMUSIC
```

としてZ-MUSICを常駐させてください。そうそう，これは対戦ゲームなので，対戦相手も調達してくださいね。

あとは，いつもどおりBASICを立ち上げて，リストを打ち込んで，RUNすればゲームスタート！

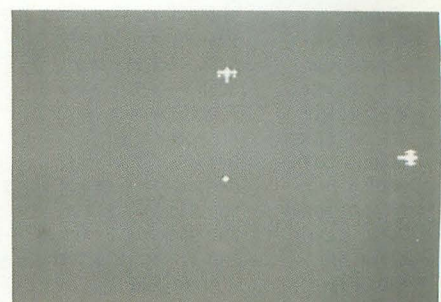
プレイヤー1，2はそれぞれジョイスティック1，2で赤タンク，青タンクを操って敵のタンクを撃ってください。タイマの256カウントが０になるか，プレイヤー１または２が敵を15回撃てば勝ちです。タンクの移動はジョイスティックの上下左右斜めが，そのままタンクの移動方向で，Ａボタンで弾が発射されます。

実はこのリストは，ちょっくらいじって掲載してしまいました。だって，最初はタイマが1000もあったので，ゲームが終わるまでにプレイヤーの２人とも疲れてしまうし，しかも何発相手に当てても時間まで終わらないんだもん。さながらタイマの切れたエアホッケーのようですよ。ゼエゼへ。

あ，改造は非常に簡単でした。リストの40行でtime＝256と書いてあるところがゲームスタート時点のタイマの残り，それから475行のif((atari1 ＝> 15))……というのがどちらかが15点以上取ったかどうかの処理です。

なお，タイマは厳密に時間計測をしていないので速いマシンだとちゃっちゃか終わってしまいます。自分のマシンに合わせてtimeの値を変えてみてください。リスト自体は見やすいので，皆さんも「こうしたいんだ」というのがありましたら自分でいろいろ書き換えてみてくださいね。

ゼンジーの便利なツールあるね

さて，２本目は……あ，どこかで見たような人だ(笑)。神奈川県の西川善司さんの作品で，環境変数設定ユーティリティ，

BATTLE.BAS

DSR.Xです。どうぞっ。

DSR.X for X680x0
(要アセンブラ，リンカ)
神奈川県　西川　善司

プログラムはアセンブラのソースリストの形で書かれています。エディタでリスト2を入力し，アセンブル，リンク作業を行ってDSR.Xという実行ファイルを作ってください。

このプログラムはドライブ情報を環境変数に反映するプログラムです。現在接続されている装置を検索し，その装置名に対応したドライブ番号を環境変数に反映します。たとえば2HDフロッピーディスクが2基，ハードディスクが1基（SCSI-IDが3だとする）接続されていて，それぞれドライブA，B，Cだとします。このときDSR.Xを実行すると，

　@2HD0＝A:
　@2HD1＝B:
　@SCSI3_HD0＝C:

というような環境変数が設定されます。装置の種類を表す環境変数名は表1のとおりで，環境変数はすべて大文字です。表1のsにはSCSI-ID番号(0,1,2,……7)が入り，nにはユニット番号(0,1,2,……)が入ります。

表1　装置の種類

環境変数名	装置
@2HDn	2HDフロッピーディスクドライブ
@RAMDISKn	RAMディスク
@SASI_HDn	SASIハードディスク
@SCSIs_HDn	SCSIハードディスク
@SCSIs_MOn	光磁気ディスク
@SCSIs_CDROMn	CD-ROMドライブ
@SCSIs_DATn	DATドライブ
@2HCn_2HC	2HCフロッピーディスクドライブ
@@640_2DDn	2DD 640Kバイトフロッピーディスクドライブ
@720_2DDn	2DD 720Kバイトフロッピーディスクドライブ
@144_2HD	2HD 1.44Mバイトフロッピーディスクドライブ
@OTHERn	その他の装置

たとえば，X68000を起動したときにSASIハードディスクがブートデバイスであった場合，ハードディスクのドライブ名がA:になりますね。このときにDSR.Xをコマンドラインや AUTOEXEC.BATで実行すると，

　@SASI_HD0＝A:

という環境変数ができます。また，SCSI-ID0のMOから起動したときにはMOがドライブA:になるので，

　@SCSI0_MO0＝A:

となるわけです。さらにSCSI-ID3,4にMOが接続されているとします。そして，SCSI-ID3のMOは2つのパーティションに切られていて，さらにRAMDISKが3つ設定されている場合は，

　@SCSI3_MO0＝B:
　@SCSI3_MO1＝C:
　@SCSI4_MO0＝D:
　@RAMDISK0＝E:
　@RAMDISK1＝F:
　@RAMDISK2＝G:

というように設定されます。

あーったく，善ちゃんってば，最近出番がちょっと減ってるなと思っていたら，こんなとこにも出てきたのね。いや，大歓迎ですけどね。

このプログラムなんですが，いろいろ応用的な使い方ができますね。たとえば，いま流行りのDOS/V機なんかがそうですけど，どんなふうにハードディスクやMOがつながっていても，フロッピードライブをドライブA:，B:にしたい，なんて場合があります。こんなときには，DSR.XをAUTOEXEC.BATの先頭で実行しておき，さらに，DRIVE.Xコマンドで，

　DRIVE %@2HD0% A: > NUL
　DRIVE %@2HD1% B: > NUL
　（> NUL は単なる表示抑制）

とすればいかなる状況でもFDDがドライブA，Bに設定できますね。

ちょっと使い方がややこしい気もするけど，自由度の高いプログラム，さすが善ちゃんですね。また，このプログラムはSASIインタフェイスをSCSIインタフェイスの代わりに使うためのソフトウェア，SxSIにも対応しているんだそうで，うーん，たいしたもんだわ。私も見習わなくちゃだな，うんうん。

あ，そうだ。なお，配布は自由，改変も自由ですので，どんどん改良してネットに流しちゃいましょう。ライセンスも放棄しますので無断で商用に使うのも許可します。ただし責任も放棄します，とのことですのでそのように扱ってくださいね。

チューブを走れ！ 緑八角！

ふうっ，真面目なユーティリティの解説は疲れます。それでは，今月最後のプログラムはゲームプログラムでまいりましょう。この人もホントに押しも押されもしない常連になりましたね。平井さんのプログラムでOCT1.BASです。どうぞっ。

OCT1.BAS for X680x0（要Cコンパイラ）
三重県　平井　栄治

このOCT1.BASはX-BASICソースファイルですが，ニーモニックが含まれているのでインタプリタ上では入力も実行もできません。コンパイル専用です。エディタでこのリストを入力してから，オプティマイザを外し，以下のようにコンパイルして，

　A>CC /Y OCT1.BAS

OCT1.Xという実行ファイルを作ってください。さて，オブジェクトが作れたら遊び方です。

まず，“ESC or s”と表示されているときSキーを押すと遊戯開始です。マイキャラ緑八角形(だってそうなんだもん)は，青のチューブの中を進んでいきます。緑八角形をカーソルキーで左右に動かし，隙間を避けて進んでください。チューブを1コマ進むごとにスコアが2点増加します。16コマ進むごとに出現する赤八角形を取ると，ボーナスで得点が8点増加します。25個ある赤八角形が出尽くすか，緑八角形がチューブの隙間を踏んでしまうとゲーム終了です。

また，遊戯中にSキーを押すと遊戯を中断して“ESC or s”と表示し，ESCキーを押すと親プロセスに戻ります。

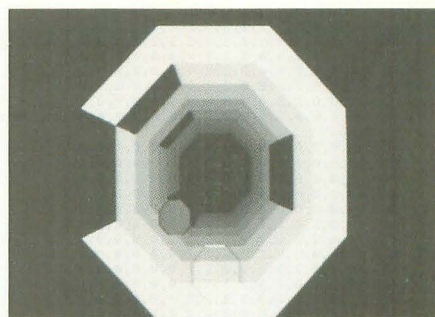
ということなんですが，あ，なんか某ソニック&テイルズっぽいですね。要するにチューブの中を進んでいくわけですが，ちょっと，画面を見ているだけだと穴が障害だっていうのがわかりにくいかな。失敗したときに緑八角が「ひゅ〜」とでも落ちていくとわかりやすかったかも。まだちょっと改良の余地ありかもしれません。

なおこのプログラムは垂直同期による割り込みが使用されている場合は実行できません。それから，適当なオプションをつけ

るとゲームの実行スピードが速くなります。なんかあんまりゲームにならんような気がするんですけどもね。

でも，赤八角形が隙間上にあると取った時点で遊戯終了しちゃいますし，前後が隙間だと取りにくくなるので，それらの位置には隙間が作成されないようにしてあったりして，押さえるところは押さえています。でも，平井さんは常連さんだからね，厳しいかもしれないけど75点だな。

そうそう，あとね，平井さんのリストっていつも真四角になってて読みづらいんです。お願いですから，もう少し読みやすくしてください。(横幅はちょうどいいんだけど）私は読むだけだからいいんだけど，実際に整形しなおす編集さんは泣いてますんで。もちろんそれ以外の皆さんもなるべく読みやすいリストでよろしくね。

さて，今月はこれにておしまい。また来月お会いしましょう。

OCTI.BAS

リスト1 BATTLE.BAS

```
10 key 6,"screen 1,3,1,1"+chr$(13):color 7
20 screen 0,1,1,1:sp_init():sp_xinit():sp_disp(1)
30 int x1,y1,x2,y2,p1,p2,j1=2,j2=6,sw1,sw2,stat1,stat2
40 int hit1,hit2,atari1,atari2,loop1,loop2,time=256
50 sprite_pat():sound()
60 vpage(0)
70 apage(0):fill(16,32,256,256,1)
80 apage(1):fill(0,0,256,16,2):fill(0,17,256,256,3)
90 palet(2,45000):palet(3,29000):palet(4,1200):palet(5,5000)
100 for i=0 to 7:sp_set(2,,,&H100+i,3):next
110 sp_set(2,,,&H102,3):sp_set(0,,,&H108,3)
120 for j=0 to 7:sp_set(3,,,&H200+j,3):next
130 sp_set(3,,,&H206,3):sp_set(1,,,&H208,3)
140 sp_loc(2,32,144):sp_loc(3,240,144):sp_off(2,3)
150 sp_loc(0):sp_loc(1)
160 sp_slon(0)
170 sp_hgadd(0,0):sp_hgadd(1,1)
180 for i=2 to 3:sp_hitrng(i,8,8,8,8):next
190 sp_hiton(2,1):sp_hiton(3,0)
200 while 1
210 sp_stkoff(1):sp_stkoff(2)
220 apage(2):fill(0,0,256,256,4)
230 moji(35,10,"来るなら来いっ!",1,2)
240 moji(0,60,"ﾊﾞﾄﾙ ﾌｨｰﾙﾄﾞ",2,6)
250 moji(60,200,"PUSH A START",1,1)
260 moji(60,230,"PUSH B  END ",1,1)
270 vpage(4):cls:sp_on(2,3):palet(15,2000)
280 sp_set(2,,,&H102,3):sp_set(3,,,&H206,3)
290 sp_loc(2,32,144):sp_loc(3,240,144)
300 while loop1<>1
310 sw1=xstrig(1,0)
320 switch sw1:case 1:loop1=1:break:endswitch
330 sw1=xstrig(1,1)
340 switch sw1:case 1:sp_sloff():end:break:endswitch
350 sw2=xstrig(2,0)
360 switch sw2:case 1:loop1=1:break:endswitch
370 sw2=xstrig(2,1)
380 switch sw2:case 1:sp_sloff():end:break:endswitch
390 endwhile
400 locate 1,0:print "HIT  0"
410 locate 25,0:print "HIT  0"
420 locate 12,0:print "TIME1000"
430 vpage(2)
440 sp_stkon(1,2,2,3):sp_stkon(2,3,2,3)
450 while loop2<>1
460 locate 12,0:print using"TIME####";time
470 if time=0 then loop2=1:m_play(6)
475 if((atari1 >= 15) or (atari2 >= 15 )) then loop2=1:m_play(6)
480 if time>=0 then time=time-1
490 sw1=xstrig(1,0):switch sw1:case 1:tama1():break
500 endswitch
510 sw2=xstrig(2,0):switch sw2:case 1:tama2():break
520 endswitch
530 pat1():pat2():jet1():jet2()
540 hit1=sp_hit(2)
550 if hit1=1 then {
560 m_play(7)
570 sp_loc(1,0,0)
580 sp_off(1):atari1=atari1+1
590 locate 25,0:print using"HIT###";atari1
600 sp_hiton(2,1) }
610 hit2=sp_hit(3)
620 if hit2=0 then {
630 m_play(8)
640 sp_loc(0,0,0)
650 sp_off(0):atari2=atari2+1
660 locate  1,0:print using"HIT###";atari2
670 sp_hiton(3,0) }
680 endwhile
690 apage(3)
700 sp_off(0,1)
710 if atari1<atari2 then win1() else {
720 if atari1>atari2 then win2() else {
730 draw() }}
740 vpage(8)
750 loop1=0:loop2=0
760 wait(30000)
770 atari1=0:atari2=0
780 j1=2:j2=6
790 sw1=0:sw2=0
800 time=1000
810 endwhile
820 end /*********************************
830 func draw()
840 palet(5,4000)
850 fill(0,0,256,256,5)
860 palet(15,62000)
870 moji(30,30,"引き分け",2,8)
880 endfunc
890 func win1()
900 palet(6,1600)
910 fill(0,0,256,256,6)
920 palet(15,2000)
930 moji(10,30,"赤戦車勝利",2,8)
940 endfunc
950 func win2()
960 palet(7,35000)
970 fill(0,0,256,256,7)
980 palet(15,3000)
990 moji(10,30,"青戦車勝利",2,8)
1000 endfunc
1010 func moji(x;int,y;int,m;str,a;char,b;char)
1020 symbol(x,y,m,a,b,2,15,0)
1030 endfunc
1040 func pat1()
1050 p1=stick(1)
1060 switch p1
1070 case 8:j1=0:break
1080 case 9:j1=1:break
1090 case 6:j1=2:break
1100 case 3:j1=3:break
1110 case 2:j1=4:break
1120 case 1:j1=5:break
1130 case 4:j1=6:break
1140 case 7:j1=7:break
1150 endswitch
1160 endfunc
1170 func pat2()
1180 p2=stick(2)
1190 switch p2
1200 case 8:j2=0:break
1210 case 9:j2=1:break
1220 case 6:j2=2:break
1230 case 3:j2=3:break
1240 case 2:j2=4:break
1250 case 1:j2=5:break
1260 case 4:j2=6:break
1270 case 7:j2=7:break
1280 endswitch
1290 endfunc
1300 func jet1()
1310 switch j1
1320 case 0:sp_set(2,,,&H100,3):x1=   0:y1=-240:break
1330 case 1:sp_set(2,,,&H101,3):x1= 240:y1=-240:break
1340 case 2:sp_set(2,,,&H102,3):x1= 240:y1=   0:break
1350 case 3:sp_set(2,,,&H103,3):x1= 240:y1= 240:break
1360 case 4:sp_set(2,,,&H104,3):x1=   0:y1= 240:break
1370 case 5:sp_set(2,,,&H105,3):x1=-240:y1= 240:break
1380 case 6:sp_set(2,,,&H106,3):x1=-240:y1=   0:break
1390 case 7:sp_set(2,,,&H107,3):x1=-240:y1=-240:break
1400 endswitch
1410 endfunc
1420 func jet2()
1430 switch j2
1440 case 0:sp_set(3,,,&H200,3):x2=   0:y2=-240:break
1450 case 1:sp_set(3,,,&H201,3):x2= 240:y2=-240:break
1460 case 2:sp_set(3,,,&H202,3):x2= 240:y2=   0:break
1470 case 3:sp_set(3,,,&H203,3):x2= 240:y2= 240:break
1480 case 4:sp_set(3,,,&H204,3):x2=   0:y2= 240:break
1490 case 5:sp_set(3,,,&H205,3):x2=-240:y2= 240:break
1500 case 6:sp_set(3,,,&H206,3):x2=-240:y2=   0:break
1510 case 7:sp_set(3,,,&H207,3):x2=-240:y2=-240:break
1520 endswitch
1530 endfunc
1540 func tama1()
1550 stat1=sp_xstat(0)
1560 if stat1=0 then {
1570 sp_on(0):sp_loc(0,sp_stat(2,0),sp_stat(2,1))
1580 sp_slidev(0,x1,y1,30) }
1590 endfunc
1600 func tama2()
1610 stat2=sp_xstat(1)
1620 if stat2=0 then {
1630 sp_on(1):sp_loc(1,sp_stat(3,0),sp_stat(3,1))
1640 sp_slidev(1,x2,y2,30) }
1650 endfunc
1660 func wait(wa;int):for w=0 to wa:next:endfunc
1670 func sprite_pat()
1680 dim int pal(15)
1690 pal={
1700 +0,65473,1985,63,1,1,1,1,1,1.1,1,1,1,1,65535}:for k=0 to 15;sp_c
olor(k,pal(k),1):next
1710 pal={
1720 +0,1984,19070,63504,0,0,0,0,0,0,0,0,0,1,1}:for k=0 to 15:sp_co
lor(k,pal(k),2):next
1730 dim char sp(255)
1740 sp={
1750 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1760 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1770 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,
1780 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,
1790 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,
1800 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,
1810 +0,0,0,1,0,0,0,2,2,0,0,0,1,0,0,0,
1820 +0,0,0,2,0,0,2,2,2,2,0,0,2,0,0,0,
1830 +0,0,0,2,0,0,2,3,3,2,0,0,2,0,0,0,
```

```
1840 +0,0,2,2,2,2,2,3,3,2,2,2,2,2,0,0,
1850 +0,0,2,2,2,2,2,3,3,2,2,2,2,2,0,0,
1860 +0,0,0,2,0,0,2,2,2,2,0,0,2,0,0,0,
1870 +0,0,0,2,0,0,0,2,2,0,0,0,2,0,0,0,
1880 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,
1890 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1900 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(0,sp,1)
1910 sp={
1920 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1930 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1940 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1950 +0,0,0,0,0,1,0,0,0,0,0,2,0,0,0,0,
1960 +0,0,0,0,2,0,0,0,0,0,2,2,2,0,0,0,
1970 +0,0,2,2,0,0,0,0,0,2,2,2,0,0,0,0,
1980 +0,0,2,2,2,2,0,0,2,2,2,0,0,0,0,0,
1990 +0,2,0,2,2,2,2,2,2,2,0,0,0,0,0,0,
2000 +0,0,0,2,2,2,3,3,2,0,0,0,0,0,0,0,
2010 +0,0,0,0,2,3,3,3,2,0,0,0,0,0,0,0,
2020 +0,0,0,0,2,3,3,2,2,2,0,0,1,0,0,0,
2030 +0,0,0,2,2,2,2,2,2,2,0,2,0,0,0,0,
2040 +0,0,0,0,2,0,0,2,2,2,2,0,0,0,0,0,
2050 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2060 +0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,
2070 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(1,sp,1)
2080 sp={
2090 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2100 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2110 +0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,
2120 +0,0,0,2,2,2,2,2,2,1,0,0,0,0,0,0,
2130 +0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,
2140 +0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,
2150 +0,0,0,0,2,2,2,2,2,0,0,0,0,0,0,0,
2160 +0,0,2,2,2,3,3,3,2,2,2,2,2,2,0,0,
2170 +0,0,2,2,2,3,3,3,2,2,2,2,2,2,0,0,
2180 +0,0,0,0,2,2,2,2,2,0,0,0,0,0,0,0,
2190 +0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,
2200 +0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,
2210 +0,0,0,2,2,2,2,2,2,1,0,0,0,0,0,0,
2220 +0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,
2230 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2240 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(2,sp,1)
2250 sp={
2260 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2270 +0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,
2280 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2290 +0,0,0,0,2,0,0,2,2,2,2,0,0,0,0,0,
2300 +0,0,0,2,2,2,2,2,2,2,0,2,0,0,0,0,
2310 +0,0,0,0,2,3,3,2,2,2,0,0,1,0,0,0,
2320 +0,0,0,0,2,3,3,3,2,0,0,0,0,0,0,0,
2330 +0,0,0,2,2,2,3,3,2,0,0,0,0,0,0,0,
2340 +0,2,0,2,2,2,2,2,2,2,0,0,0,0,0,0,
2350 +0,0,2,2,2,2,0,0,2,2,2,0,0,0,0,0,
2360 +0,0,2,2,0,0,0,0,0,2,2,2,0,0,0,0,
2370 +0,0,0,0,2,0,0,0,0,0,2,2,2,0,0,0,
2380 +0,0,0,0,0,1,0,0,0,0,0,2,0,0,0,0,
2390 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2400 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2410 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(3,sp,1)
2420 sp={
2430 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2440 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2450 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,
2460 +0,0,0,2,0,0,0,2,2,0,0,0,2,0,0,0,
2470 +0,0,0,2,0,0,2,2,2,2,0,0,2,0,0,0,
2480 +0,0,2,2,2,2,2,3,3,2,2,2,2,2,0,0,
2490 +0,0,2,2,2,2,2,3,3,2,2,2,2,2,0,0,
2500 +0,0,0,2,0,0,2,3,3,2,0,0,2,0,0,0,
2510 +0,0,0,2,0,0,2,2,2,2,0,0,2,0,0,0,
2520 +0,0,0,1,0,0,0,2,2,0,0,0,1,0,0,0,
2530 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,
2540 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,
2550 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,
2560 +0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,
2570 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2580 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(4,sp,1)
2590 sp={
2600 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2610 +0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,
2620 +0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,
2630 +0,0,0,0,0,2,2,2,2,0,0,2,0,0,0,0,
2640 +0,0,0,0,2,0,2,2,2,2,2,2,2,0,0,0,
2650 +0,0,0,1,0,0,2,2,2,3,3,2,0,0,0,0,
2660 +0,0,0,0,0,0,0,2,3,3,3,2,0,0,0,0,
2670 +0,0,0,0,0,0,0,2,3,3,2,2,2,0,0,0,
2680 +0,0,0,0,0,0,2,2,2,2,2,2,2,0,2,0,
2690 +0,0,0,0,0,2,2,2,0,0,2,2,2,2,0,0,
2700 +0,0,0,0,2,2,2,0,0,0,0,2,2,0,0,
2710 +0,0,0,2,2,2,0,0,0,0,2,0,0,0,0,
2720 +0,0,0,0,2,0,0,0,0,1,0,0,0,0,0,
2730 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2740 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2750 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(5,sp,1)
2760 sp={
2770 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2780 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2790 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2800 +0,0,0,0,0,0,1,2,2,2,2,2,2,0,0,0,
2810 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2820 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2830 +0,0,0,0,0,0,0,2,2,2,2,2,0,0,0,0,
2840 +0,0,2,2,2,2,2,2,3,3,3,2,2,2,0,0,
2850 +0,0,2,2,2,2,2,2,3,3,3,2,2,2,0,0,
2860 +0,0,0,0,0,0,0,2,2,2,2,2,0,0,0,0,
2870 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2880 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2890 +0,0,0,0,0,0,1,2,2,2,2,2,2,0,0,0,
2900 +0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,
2910 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2920 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(6,sp,1)
2930 sp={
2940 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2950 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2960 +0,0,0,2,0,0,0,0,0,1,0,0,0,0,0,0,
2970 +0,0,2,2,2,0,0,0,0,0,2,0,0,0,0,0,
2980 +0,0,0,2,2,2,0,0,0,0,0,2,2,0,0,0,
2990 +0,0,0,0,2,2,2,0,0,2,2,2,2,0,0,0,
3000 +0,0,0,0,0,2,2,2,2,2,2,2,0,2,0,0,
3010 +0,0,0,0,0,0,2,3,3,2,2,2,0,0,0,0,
3020 +0,0,0,0,0,0,2,3,3,3,2,0,0,0,0,0,
3030 +0,0,1,0,0,2,2,2,3,3,2,0,0,0,0,0,
3040 +0,0,0,2,0,2,2,2,2,2,2,2,0,0,0,0,
3050 +0,0,0,0,2,2,2,2,0,0,2,0,0,0,0,0,
3060 +0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,0,
3070 +0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,
3080 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3090 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(7,sp,1)
3100 sp={
3110 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3120 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3130 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3140 +0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,
3150 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3160 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3170 +0,0,0,0,0,4,2,2,0,0,0,0,0,0,0,0,
3180 +0,0,0,0,0,2,2,2,2,0,0,0,0,0,0,0,
3190 +0,0,0,0,0,2,2,2,2,0,0,0,0,0,0,0,
3200 +0,0,0,0,0,0,2,2,4,0,0,0,0,0,0,0,
3210 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3220 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3230 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3240 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3250 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3260 +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}:sp_def(8,sp,1)
3270 endfunc
3280 func sound()
3290 dim char bakuhatu1(4,10)={
3300  7,15, 3, 0,255,69, 0, 6, 2, 3, 0,
3310  8,16,10, 8,  8, 0, 0, 3, 2, 2, 1,
3320  8, 9,31,14, 15, 5, 0, 6, 2, 2, 1,
3330  0,14,12, 7,  5, 0, 0, 5, 3, 1, 1,
3340 31, 9,10,14,  4, 0, 0, 7, 4, 1, 1}
3350 m_vset(80,bakuhatu1)
3360 for m=6 to 8:m_alloc(m,19):m_assign(m,m):next
3370 m_trk(6,"v15@80o3l8(c<c)&(c>c)")
3380 m_trk(7,"v15@80o4c1")
3390 m_trk(8,"v15@80o4e1")
3400 endfunc
```

リスト2　DSR.S

```
 1: *           DSR.R V1.2
 2: *
 3: *           PROGRAMMED BY Z.NISHIKAWA
 4:
 5:             include iocscall.mac
 6:             include doscall.mac
 7:
 8: unit:               equ     1
 9: devptr:             equ     18          *デバイスドライバまでのオフセット値
10: media:              equ     22          *メディアバイトまでのオフセット値
11:
12:             lea     USERSP(pc),sp
13:
14:             pea     TITLE(pc)
15:             DOS     _PRINT
16:             addq.w  #4,sp
17:
18:             lea     work(pc),a6
19:
20:             clr.l   -(sp)
21:             DOS     _SUPER
22:             addq.w  #4,sp
23:             move.l  d0,SSPBUF-work(a6)
24:
25:             moveq.l #1,d2               *0 is current
26:             lea     DPBbuff(pc),a5
27: mlp00:
28:             pea.l   (a5)
29:             move.w  d2,-(a7)
30:             DOS     _GETDPB
31:             addq.l  #6,a7
32:             tst.l   d0
33:             bmi     all_end?
34:
35:             move.b  media(a5),d1
36:
37:             lea     _2HD(pc),a0
38:             cmpi.b  #$fe,d1             *2HD
39:             beq     st_type
40:
41:             lea     RAMDISK(pc),a0
42:             cmpi.b  #$f9,d1             *RAMDISK
43:             beq     st_type
44:
45:             lea     SASI_HD(pc),a0
46:             cmpi.b  #$f8.d1             *SASI HD
47:             beq     st_type
48:
49:             lea     SCSI_HD(pc),a0
50:             cmpi.b  #$f7,d1             *SCSI HD
51:             beq     st_scsi_id
52:
53:             lea     MO(pc),a0
54:             cmpi.b  #$f6,d1             *MO
55:             beq     st_scsi_id
56:
57:             lea     _2HC(pc),a0
58:             cmpi.b  #$fd,d1             *2HC
59:             beq     st_type
60:
61:             lea     _640_2DD(pc),a0
62:             cmpi.b  #$fb,d1             *2DD 640
63:             beq     st_type
64:
65:             lea     _720_2DD(pc),a0
66:             cmpi.b  #$fc,d1             *2DD 720
```


▶必要に迫られてポケベルをもつようになった。おかげで夜の外出が多くなり，X68000の相手をする時間がなくなってしまった。僕のX68000はいまごろなにをしているのだろう。

小山　優一(21)東京都

```
 67:            beq     st_type
 68:
 69:            lea     _144_2HD(pc),a0
 70:            cmpi.b  #$fa,d1             *2HD 144
 71:            beq     st_type
 72:
 73:            lea     CDROM(pc),a0
 74:            cmpi.b  #$f5,d1             *CDROM
 75:            beq     st_scsi_id
 76:
 77:            lea     DAT(pc),a0
 78:            cmpi.b  #$f4,d1             *DAT
 79:            beq     st_scsi_id
 80:
 81:            lea     OTHER(pc),a0        *OTHER
 82:            bra     st_type
 83: st_scsi_id:
 84:            bsr     get_scsi_id
 85:            add.b   #'0',d0
 86:            move.b  d0,5(a0)
 87: st_type:
 88:            lea     ENVbuff(pc),a1
 89: stlp:
 90:            move.b  (a0)+,(a1)+
 91:            bne     stlp
 92: AA:
 93:            move.b  unit(a5),d1
 94:            add.b   #$30,d1
 95:            move.b  d1,-1(a1)
 96:            clr.b   (a1)+               *ENV name completed
 97:
 98:            move.b  d2,d1
 99:            add.b   #$40,d1             *A B C...
100:            move.b  d1,DRVno-work(a6)
101:
102:            pea     DRVno(pc)
103:            clr.l   -(sp)
104:            pea     ENVbuff(pc)
105:            DOS     _SETENV
106:            addq.w  #8,sp
107:
108:            addq.w  #1,d2
109:            bra     mlp00
110:
111: all_end?:
112:            addq.w  #1,d2
113:            cmpi.b  #26,d2
114:            bls     mlp00
115:
116:            move.l  SSPBUF(pc),-(sp)
117:            DOS     _SUPER
118:            addq.w  #4,sp
119:
120:            DOS     _EXIT
121:
122: get_scsi_id:
123:            * < a5.l=DPBbuff
124:            * > d0.b=SCSI ID
125:            * X a2
126:            move.l  devptr(a5),a2
127:            cmpi.l  #'Hero',$18(a2)             *check SxSI ID
128:            bne     gsi2
129:            move.b  $28(a2),d0
130:            rts
131: gsi2:
132:            add.w   $74(a2),a2
133:            move.w  $61e(a2),d0
134:            rts
135:
136: work:
137: TITLE:              dc.b    'DRIVE STATUS REFLECTOR ver.1.2 (C) 19
94 ZENJI SOFT',13,10,0
138: _2HD:               dc.b    '@2HD',0
139: RAMDISK:            dc.b    '@RAMDISK',0
140: SASI_HD:            dc.b    '@SASI_HD',0
141: SCSI_HD:            dc.b    '@SCSIx_HD',0
142: MO:                 dc.b    '@SCSIx_MO',0
143: CDROM:              dc.b    '@SCSIx_CDROM',0
144: DAT:                dc.b    '@SCSIx_DAT',0
145: _2HC:               dc.b    '@2HC',0
146: _640_2DD:           dc.b    '@640_2DD',0
147: _720_2DD:           dc.b    '@720_2DD',0
148: _144_2HD:           dc.b    '@144_2HD',0
149: OTHER:              dc.b    '@OTHER',0
150: DRVno:              dc.b    ' :',0
151:            .even
152:            .bss
153: DPBbuff:            ds.b    94
154: ENVbuff:            ds.b    256
155: SSPBUF:             ds.l    1
156:                     ds.l    256
157: USERSP:
```

リスト3 OCT1.BAS

```
  10 int h,i,j,k,m,n,o,p,q,r,s,t,u=264,w=  1:dim char y(  7)={
  20  69,83,67,32,111,114,32,83}:v=  1:x=  1:dim char z(391)
  30 #c
  40 #asm
  50          lea.l   vdisp0,a2          *割り込み処理アドレスを
  60          move.l  a2,_h              *得る
  70 #endasm
  80    h=VDISPST(h,0,1);
  90 #endc
 100 if h then{}else{if b_argc=1 then{b_argc=15}else{b_argc=14
 110   }:screen 0,2,1,1:sp_init():palet(65,19026):sp_disp(1)
 120 #c
 130    TPALET( 3,19026);B_SUPER(0);
 140 #asm
 150          move.w  #$044,$e80000    *水平トータル
 160          move.w  #$006,$e80002    *水平同期終了位置
 170          move.w  #$013,$e80004    *水平表示開始位置
 180          move.w  #$033,$e80006    *水平表示終了位置
 190          move.w  #$237,$e80008    *垂直トータル
 200          move.w  #$005,$e8000a    *垂直同期終了位置
 210          move.w  #$028,$e8000c    *垂直表示開始位置
 220          move.w  #$228,$e8000e    *垂直表示終了位置
 230          move.w  #$01b,$e80010    *外部同期水平アジャスト
 240          move.w  #$111,$e80028    *CRTC R20
 250          move.w  #$0ff,$eb080a    *H-TOTAL
 260          move.w  #$017,$eb080c    *H-DISP
 270          move.w  #$028,$eb080e    *V-DISP
 280          move.w  #$010,$eb0810    *画面モードレジスタ
 290          ori.b   #2,$e8e007       *ドットクロック切り替え
 300 #endasm
 310 #endc
 320   sp_color(9,63488):for t=0 to 7
 330   s=417*sin(23*pi()/180)/((21-t-t)*cos(23*pi()/180))
 340      a(23,44,0,s,0):a(67,46,45,s,0):b(1):h= 0
 350   i=417*cos(23*pi()/180)/(10.5#-t)-s*cos(23*pi()/180)
 360   for j=0 to 7:h=h+45:z(1+u)=120+i*sin(h*pi()/180)
 370  z(u)=120+i*cos(h*pi()/180):u= u+2:next
 380 next:a(23,44,0,s,0):a(67,46,45,s,0)
 390 b(0):b(2):w= 1:for t=0 to 7:x=11-t
 400   a(23,44,0,417,1):w=w-7:a(67,46,45,417,1):w= w-1:next
 410   locate 12,8:print"Score000":locate 12,6:print"  High  "
 420   locate 12,7:print"score000":_(0,z(378),z(379),329):q=-1
 430    r=0:w=0:apage(1):sp_on(0,8):for h=0 to  2:for i=0 to 3
 440 symbol(112+8*i,144,chr$(240+h)+chr$(y(h+h+i)),1,1,1,65,0)
 450 next:h=h+1:next:while q<26:for h=0 to 15: z(h)=6:next
 460 p=3:s=0:t=0:u=15:v=0:x=0:locate 14,9:print"     ":vpage(3)
 470 while 0>q:b(3):if h then{q=26}else{if i then{
 480 q=0:while   i:b(3):endwhile:x=12:b(2):vpage(1)
 490 locate 14,9:print"OCT1"}else{}}:endwhile:while q<26
 500 #c
 510    h=40&BITSNS(7);
 520 #endc
 530 switch        h:        case 32:h=1:    break:case 8:h= -1
 540          break:case 0:case 40:h=0:endswitch:for i=1 to 2
 550 switch stick(i):case 1:case  4:                case 7:h=h-1
 560          break:case 3:case  6:                case 9:h=h+1
 570 endswitch: next:      h=sgn(h): if h=t then{h=0}else{t=h}
 580    if p then{i=0:p=p-1}else{i=1:p=3:s=s+2:u=u+1:switch u
 590           case  8:                     q=q+1:    v=7*rnd( )+1
 600                    if v<z(8-w) then{}else{   v=          v+1
 610    }:      case  9:                          z(w)=7*rnd( )+1
 620                    if v>z(  w) then{}else{z(w)=     z(w)+1
 630    }:break:case 16:       u=0:v=0:default:z(w)=8*rnd( )+1
 640 endswitch}:  for j=i to 7:o=z(8-i+j-w):z(j+w)=b(4):  next
 650                if    i then{o=z(       w):z(  w)=b(4)}else{}
 660                if    v then{o=           v:     v=b(4)}else{}
 670  if x>b_argc then{h=x+1}else{h=b_argc}:while h>x:endwhile
 680 for h=0 to 7:i=h*16+2*z(8+h-w):j=h*16+2*z(h+w):k=z(256+h)
 690 #c
 700 #asm
 710          lea.l   $e82000,a2        *グラフィック用パレット
 720          move.l  _i,d2             *インデックス
 730          move.l  _j,d1             *インデックス
 740          move.l  _k,d0             *色データ
 750          move.w  d0,(a2,d2)        *古い隙間を削除
 760          clr.w   (a2,d1)           *新しい隙間を作成
 770 #endasm
 780 #endc
 790 next:x=12:if v then{
 800  _(9,z(134+16*u+v+v),z(135+16*u+v+v),176+9*u):sp_on(9,17)
 810    }else{sp_off(9,17)}:if u=15 and v=2 then{s=s+8:v= 0
 820     sp_color(8,63488):if s>999 then{p=0:q=25:s=999:u=15
 830    }else{}}else{sp_color(8,1984)}:if r<s then{r=s}else{}
 840 locate 17,7:print right$("00"+itoa(r),3)
 850 locate 17,8:print right$("00"+itoa(s),3)
 860 w=8-w:if q=25 and u=15 then{if p=0 or v=0 then{q=-1:break
 870   }else{}}else{}:if z(15-w)=2 then{q=-1:break}else{}:b(3)
 880   if h then{q=26}else{if i then{q=-1:while i:b(3)
 890   endwhile:break}else{}}:endwhile:endwhile:screen 0,2,1,1
 900 #c
 910   VDISPST(0,0,1);
 920 #asm
 930   and.b #$fd,$e8e007        *ドットクロック切り替え
 940 #endasm
 950 #endc
 960   while inkey$(0)<>"":endwhile}:end
 970 #c
 980 #asm
 990   vdisp0: cmpi.l  #12,_x           *_xが
1000           blt     vdisp1           *12未満の場合
1010           addq.l  #1,_x            *カウント
1020   vdisp1: rte                      *V-DISPによる割り込み
1030 #endasm
1040 #endc
1050 func a(g,f,e,d,c):for h=0 to 3:i=f+ g
1060      j=128+d*cos(g*pi()/180)/x:o=128+d*sin(g*pi()/180)/x
1070      k=128+d*cos(i*pi()/180)/x:p=128+d*sin(i*pi()/180)/x
1080        x= -c+x
1090      m=128+d*cos(g*pi()/180)/x:q=128+d*sin(g*pi()/180)/x
1100      n=128+d*cos(i*pi()/180)/x:r=128+d*sin(i*pi()/180)/x
1110        x=  c+x:line(j,o,k,p,w):line(k,p,n,r,w)
1120             line(j,o,m,q,w):line(m,q,n,r,w)
1130 paint(k+(c+c)*cos(e*pi()/180),p+(c+c)*sin(e*pi()/180),w)
1140        w=c+c+w:e=e+90:g=g+90:next:endfunc
1150  func b(g):switch g:case 2:h=0:for i=0 to 7
1160       j=108.5#/(10.5#-i):j=j+j:for k=0 to 7:if k=5 then{
1170        palet(1+h,0)}else{palet(1+h,j)}:z(256+h/8)=j
1180     sp_color(1+h/8,j*32):h=h+1:next:next:break:case 3
1190 #c
1200     h=2&BITSNS(0);i=128&BITSNS(3);
1210 #endc
1220  i=i+strig(1)+strig(2):break:case 4:o=o+h:if o<1 then{
1230  o=8}else{if o>8 then{o=1}else{}}:return(o):default
1240  paint(128,128,g*w):h=104:for i=0 to 2:j=104:for k=0 to 2
1250  get(h,j,15+h,15+j,z):sp_def(v,z,1):v=v+1:j=j+16
1260  next:h=h+16:next:w=w+1:wipe():endswitch:endfunc
1270  func _(g,f,e,d    ): for h=0 to 2:for i=0 to 2
1280  sp_set(g,f,e,d+g,3):g=g+ 1:e=e+16:next
1290  e=e-48:f=f+16:next:endfunc
1300 /* OCT1                 BAS
```

▶「響子 in CGわ～るど」の「都会猫軒のチョコレート」で，主人公が男性だったとは1本取られました。　渋谷　洋明(20)新潟県

SIDE A

ゼロヨンといえども奥は深い

Tan Akihiko 丹　明彦

直線運動を確認するため，ゼロヨンレースモデルを作ってみる
まだ，導入する要素も少なく，モデルとしては単純なものだ
しかし，いままでの理論を実装し，正しいかどうかを検証しなくてはならない

今月の車ゲー日記

ナムコの「エースドライバー」は，１週間ほどの練習で，EXPERT(PRO)モードでも対戦ならどうにか完走できるようになった。セガの「デイトナUSA」の上達にものすごく時間がかかったことを考えると，ややあっけない気がしなくもない。私がドライブゲーム慣れしてきたせいなのかもしれないが，ゲームの性格そのものにも原因があると思われる。

「エースドライバー」は基本的にドリフトゲームではなく，コーナーのRに合わせて車速とギアを合わせることでコーナリングするドライブゲームである。つまり正しく現代フォーミュラマシンの運転に則っている。どうしてもドリフトゲームに比べて不確定要素には欠けがちになる。

今後は邪魔っけな敵車の処理を覚え，タイムを切り詰めることになるだろう。そして大型筐体で完走すること。どうも私は椅子が動く筐体には弱い。

そしてDOS/Vマシン向けに，「NASCAR RACING」が発売された。「デイトナUSA」のモデルとなったNASCARシリーズのシミュレーションである。制作はお馴染み，インディカーのシミュレーションを作り続けてきたPapyrus(近作「INDYCAR RACING」)。

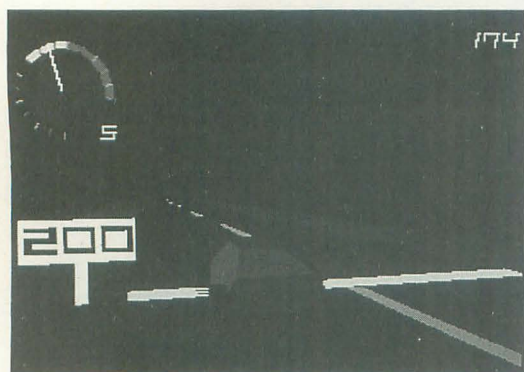
今回制作したゼロヨンレースモデル

NASCARは，とても米国的な自動車レースである。いかにもアメ車という感じの大排気量V8エンジンを積んだ極彩色の車が爆音を響かせてオーバルコースを暴れ回る。人気という点ではインディカーを上回るともい

われる。

で，「NASCAR RACING」を遊んでみた感想だが，まずアメリカンなV8サウンドを堪能できる。「デイトナUSA」の制作スタッフが魅せられたというのも納得できる。そして車が重い。ハコだから当然であるが，インディカーと比べて加速・減速・コーナリングのいずれも重さを感じる。「INDYCAR RACING」と同じアルゴリズムを使って定数だけを変えているのだろうが，よく車の違いが表れている。そして衝突の処理が「INDYCAR RACING」に比べて進歩していると感じた。インディカーと違い，NASCARは車どうしのぶつかりあいや壁との接触が半ば前提になっているので，このへんを精密に作るのは重要なポイントであったと思われる。

「INDYCAR RACING」と「NASCAR RACING」をプレイして思うのは，「日本のゲームは刺激に富んでいるし，とりあえず遊べるようにできている」ということ。シミュレーションは精密さが第一だから，面白くするためのウソを許さない。しかしそんな考え方ではアーケードゲームとして失敗する。逆に厳密すぎるシミュレーションは，コインを入れてからゲームオーバーまでの時間がすべてのアーケードゲームでは成立せず，記録や履歴が残していけるパーソナルコンピュータでこそ成立するものなのだろう。では家庭用ゲーム機ではどうか。長く遊ぶことを前提としたデザインも可能という点では有望。ただ買ってすぐには面白く遊べないという点が少々つらい。これらのシミュレーションが家庭用ゲーム機に移植されたとき，日本のゲーマーの成熟度というかオトナ度が試されるのかもしれない。

なお，「NASCAR RACING」には通常の画面解像度（320×240ピクセル）のほかに高解像度モード(640×480ピクセル)も用意されており，精緻なグラフィックを楽しめる。ただし処理量もそれなりになるので，486/66MHzでは事実上運転は無理。Pentiumを前提とできるDOS/Vの世界のレベルに

時代を感じてしまう。

ゼロヨンゲーム

ドライビングシミュレーションは，数式ベースで車の運動を計算しているが，はたして正しく計算できているかどうかが不安になる。数式が合っているように見えても，単位をひとつ間違えただけで車の運動はでたらめになってしまうからだ。

ということを避ける意味でも，前回から1次元上の運動（つまり直線運動）にレベルを下げ，エンジン出力と加速度と速度の関係をできるだけ精密に調べることにしている。クルマ雑誌のビデオ版などでは「頂上対決！　最新スポーツカー0→400mバトル」のような企画がありがちだが，直線運動のような単純な運動も結構奥が深いことを示しているといえるだろう。それに，ゼロヨンだと実車の性能がタイムというわかりやすい指標で出てくるので，計算が正しいかどうかを確かめられる。

では具体的なプログラミングに入る。今回までの到達点は，

- ・ゼロヨン専用のコース作成
- ・メーター関係の整備
- ・ギアチェンジ機構の実装
- ・実車のエンジン性能曲線とギア比の利用

といったところで，直線路をシフトアップしながら加速するという運動が一応可能になった。

専用コース

リアルな速度感を得るためには，スケールというものが大切である。コースの縮尺を間違えれば，時速200km/hで走っているはずなのに100km/hにも感じないということがままある。

教習所に通ったときの記憶をたぐって，コースの構成を図1のようにした。センターラインは20メートルおきに8メートルの白線を置く。道幅は片側6メートル。あとは飾りとして，100メートルおきに看板と横方向のラインを置いた。

そして初の試みとして，道路にテクスチャもどきをつけてみた。もちろんSLASHシステムはテクスチャマッピングをサポートしていない。ここは最小のコストで効果を得るべく，路面に小さなポリゴンを乱数を使ってばらまいた。のっぺらぼうの路面に比べれば，いくぶんは速度感が強調されるだろう。

ただ，いくらコースを精密に作り込んだところで，一定以上のフレームレートをキープできなければ速度感を得ることはできない。X68000/030での実証が極めて困難なのが残念だが，ドライブシミュレータという状況では，同じ運動でも30fpsで表現した映像と60fpsで表現した映像では違いをはっきり感じるという（一度アニメーションを作って比較してみようかと考えている）。まして20fpsを下回ってしまうと，特に手前の路面が流れるように動いてくれないので，テクスチャマップを施すだけ無駄になってしまう。たとえば180km/hで走っている車は毎秒50メートル走る。20fpsだと1フレームあたり2.5メートル車が進む計算になる。このくらいが人間の目で路面が流れているように認識できる限界だろう。

このコースの表示は簡単で，20メートルぶんを1ブロックとして，車の近くのブロックだけを並べて表示する。単調な直線路なので，使うパターンも少なくてすむ(通常路面1パターン，100メートルおきの看板のついた路面が0～400で5パターン，計6パターン)。

自動車のモデル化

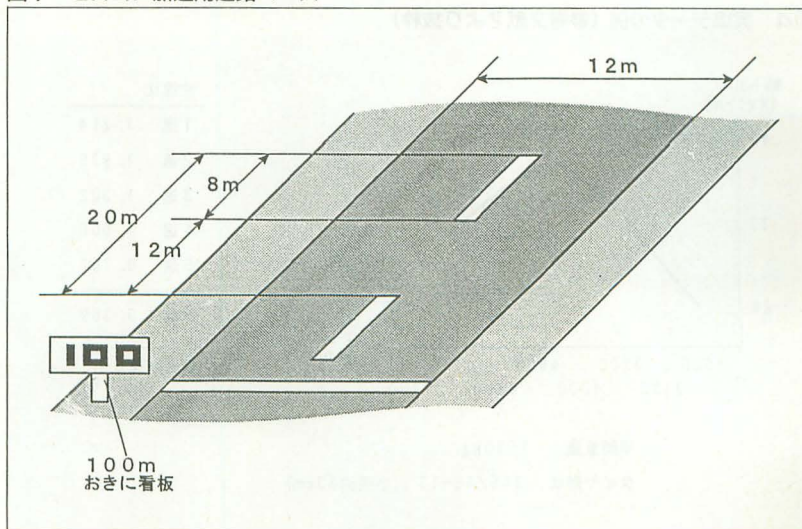
たかがゼロヨン加速といってもその運動は意外に複雑である。その要素をすべて考慮に入れてシミュレーションを行うと混乱するので，自動車についていくつかの仮定を行った（図2）。

1)　前後輪の重量配分は1：1で荷重移動なし

重心が前後の車軸の中央に位置しているものと考える。通常はエンジンレイアウトなどの関係から60：40などのように重量配分されている。また，通常はたとえば加速すると後輪により荷重がかかるものだが，今回のモデルではどんな加速（減速）をしても重量配分は変化しない。

荷重移動は有効トラクションの変化となって現れるため，FF車とFR車，そして4WD車の特性のひとつを表現する重要な手段である。

2)　路面のμは1，またタイヤは常にグリップする

図1　ゼロヨン加速用道路コース

つまり，タイヤは常に最大のトラクションを発生する。現実には路面のμが下がれば加速は鈍くなるし，それでも無理やり加速しようとすればタイヤはグリップを失いホイールスピンを始める。

グリップの消失と回復は，ゼロヨン加速をゲームとして成立させようとした場合に重要な要素となる。ロケットスタートに技量が必要になるからである。今回のモデルは，いわば完璧なトラクションコントロールのついた車である。アクセルを踏みつけていれば，車はなんのドラマもなく加速していってしまう。

3）空気抵抗なし

これは最高速に影響する要素である。エンジンから発生する駆動力と，速度の増加につれて増大する空気抵抗が釣り合う点がすなわち最高速で，ギア比などはこれをもとに決めていく。

今回は，ギア比などについては実車の値を使ったので，そのへんの心配をしなくていい。ただし最高速は不正確なものになっているだろう。レブリミットを超えるとエンジン出力が0になるように仕掛けているので，永久に加速し続けることもない。

＊　　＊　　＊

という具合なので，今回のモデルでゼロヨンタイムアタックをするのはあまり面白いものではない。差がつくとすればシフトアップのタイミングくらいのものであろう。まあ今回はタイムを計測する仕掛けも入っていないので，試作品の感はぬぐえない。

図2　車のモデル

図3　加速のプロセス

図4　実車データの例（参考文献2より抜粋）

変速比	
1速	3.214
2速	1.925
3速	1.302
4速	1.000
5速	0.752
後退	3.369
最終減速比	4.111

実装

それでは具体的なプログラムの解説に入ろう。基本的には前回の内容を素直にコーディングしただけのことである。

加速運動を計算するプロセスを図3に示す。計算は，ここしばらくお馴染みのトルク曲線をベースに行う。

1）エンジン回転数からエンジン出力を得る

エンジン出力，つまり軸トルクを，前々回で紹介した折れ線グラフライブラリの形にしておく。横軸はエンジン回転数（rpm）。

2）選択したギアの減速比を用いて駆動力を求める

前回解説したとおり，トランスミッションとは，エンジンの出力が路面（タイヤ外周）に伝わるまでの間に回転数を落とし，かわりにトルクを増加させる仕掛けである。これを求める式は次のとおり。

$$P=i\times T/r$$

P：駆動力［kgf］
i：総減速比
T：軸トルク［kgf・m］
r：タイヤの半径［m］

ギアシフトは簡単にキー入力でシフトアップ・ダウンを行う方式とした。クラッチはない。セミオートマのようなものである。

3）タイヤのトラクションの範囲内で加速する

求めた駆動力がタイヤの摩擦力を上回れば，トラ

クションコントロールでエンジン出力をカットしたと考えて，実際の駆動力はタイヤの摩擦力をいっぱいに使った値とする。求めた駆動力がタイヤの摩擦力に達しなければ，その駆動力をそのまま使う。

タイヤの摩擦力は，車の質量をm［kg］とすればμmg［N］。前後の重量配分が1：1なので各輪の摩擦力はμmg/2［N］。荷重移動がないので，この値は変わらない。また駆動形式による差もそれほど出ない（4WDはすべてのタイヤを使えるので有利）。

4） 新しい車速からエンジン回転数を求める

車速とエンジン回転数の関係は次の式。

$$n = V \times 30 \times i / (\pi \times r)$$

n：エンジン回転数［rpm］

V：車速［m/s］

ここで求めた新しいエンジン回転数を，次の瞬間の計算に用いる。以下これを繰り返す。

＊　＊　＊

今回用いた実車データは，おそれおおくも日産スカイラインGT-R(R33)である。手近にデータがあったので利用させてもらった。一部を図4に抜粋しておく。どの程度の値を用いているかの参考にしていただきたい。ちなみに総減速比iは，ギア比と最終減速比（ファイナルドライブギアの比）をかけて求める。

面白いことに，今年のGT-Rのカタログにはエンジン性能曲線が載っていないようである。昨年まではGT-Rはもちろん，GTS-tのカタログにも載っていたのに。市販車にトルクカーブはあまり意味がないというのが最近の認識なのだろうか。とはいえ，昨年末に出た三菱FTOのカタログには性能曲線が出ていたし，よくわからん。

終わりに

次回はタイヤのグリップを中心に過渡特性を盛り込んでみたい。そうすれば，今回は0/1であったアクセル開度も重要なパラメータになるだろう。そしてフルオートマのプログラミングもしてみようかな。完全コンピュータ制御の車とドラッグレースというのもいいかもしれない。

それにしてもエンジン音がないのはつまらない。連続的にピッチを変える必要があるので，X68000のAD PCMだとつらいのだ。誰か軽くて制御が簡単で，できればかっこいい音が出る効果音ドライバを持ってませんか？　周波数だけ与えればよかったMZ-80Kシリーズの発振器が懐かしい。

参考文献

1） 自動車力学，景山克三・景山一郎共著，理工図書，1984年，ISBN4-8446-0356-6

2） GOLD CARトップ　ニューカー速報No.96　NEWスカイラインGT-R，㈱交通タイムス社，1995年

■リスト　drive.c（参考）

```
  1: /*
  2:  *      drive.c
  3:  *      - 車の動作
  4:  *        Jul. 1994 - Jan. 1995  丹 明彦(Oh!X)
  5:  */
  6:
  7: ....
  8:
  9: #include        "LineGraph.h"
 10:
 11: ....
 12:
 13: LineGraph       torquecurve;
 14:                          /* 1    2    3    4    5    6    7    8    9 */
 15: double  torque_domain[] = { 0,1600,3200,4000,4400,4800,5600,7200,7400};
 16: double  torque_value[] =  {10,  20,  32,  36,  38,  36,  34,  28,   0};
 17: int     torque_nsample = 9;
 18:
 19: ....
 20:
 21: void    initCar( CarSpec *cs, CarInfo *ci )
 22: {
 23:         ....
 24:
 25: #define FINALDRIVEGEARRATIO     4.111
 26:         cs->gearratio[1] = 3.214*FINALDRIVEGEARRATIO;   /* 1st */
 27:         cs->gearratio[2] = 1.925*FINALDRIVEGEARRATIO;   /* 2nd */
 28:         cs->gearratio[3] = 1.302*FINALDRIVEGEARRATIO;   /* 3rd */
 29:         cs->gearratio[4] = 1.000*FINALDRIVEGEARRATIO;   /* 4th */
 30:         cs->gearratio[5] = 0.752*FINALDRIVEGEARRATIO;   /* 5th */
 31:         cs->gearratio[0] = 3.669*FINALDRIVEGEARRATIO;   /* reverse */
 32:         cs->nshift = 5;
 33:
 34:         torquecurve.n = torque_nsample;
 35:         torquecurve.domain = torque_domain;
 36:         torquecurve.value = torque_value;
 37:         torquecurve.inframode = LINEGRAPH_MINMAXVALUE;
 38:         torquecurve.ultramode = LINEGRAPH_MINMAXVALUE;
 39:
 40:         ....
 41: }
 42:
 43: void   drive( CarSpec *cs, CarInfo *ci )
 44: {
 45:    CarInfo   *ci0, *ci1;
 46:
 47:    ....
 48:
 49:    /* ギアシフト */
 50:    ci0->shift = ci1->shift;
 51:    if ( control.xf1key == 1 ) {
 52:       if ( ci0->shift > 1 )
 53:          ci0->shift--;
 54:    } else if ( control.xf2key == 1 ) {
 55:       if ( ci0->shift < cs->nshift )
 56:          ci0->shift++;
 57:    }
 58:
 59:    ....
 60:
 61:    /* 駆動力 */
 62:    ci1->drvf = ci1->drvr = 0.0;
 63:    if ( control.mouserb ) {
 64:       double   torque, maxforce, traction;
 65:       torque = lineGraphGetValue( &torquecurve, (double)(ci0->rpm) );
 66:       maxforce = (cs->gearratio[ci1->shift]*torque/(cs->rradius)) KGF;
 67:       traction = (MASS KGF)/2.0;
 68:       if ( cs->frontdrive ) {
 69:          if ( maxforce > traction ) {
 70:             maxforce -= traction;
 71:             ci1->drvf += traction;
 72:             ci1->pitch += -PITCH;    /* にせピッチ */
 73:          } else {
 74:             ci1->drvf += maxforce;
 75:             ci1->pitch += -(PITCH*maxforce/traction); /* にせピッチ */
 76:             maxforce = 0;
 77:          }
 78:       }
 79:       if ( cs->reardrive ) {
 80:          if ( maxforce > traction ) {
 81:             maxforce -= traction;
 82:             ci1->drvr += traction;
 83:          } else {
 84:             ci1->drvr += maxforce;
 85:             maxforce = 0;
 86:          }
 87:       }
 88:    }
 89:
 90:    ....
 91:
 92:    /* 新しい回転数 */
 93:    /*ci1->rpm = (int)((ci1->ve)*25.0 _KMPH);*/
 94:    if ( (ci1->revdf) != -1 ) {
 95:       ci1->rpm = (int)(((ci1->ve)*30.0*(cs->gearratio[ci1->shift]))/
 96:             (M_PI*(cs->rradius)));
 97:    } else {
 98:       ci1->rpm = -(int)(((ci1->ve)*30.0*(cs->gearratio[0]))/
 99:              (M_PI*(cs->rradius)));
100:    }
101:    /* レブリミット */
102:    if ( (ci1->rpm) > 7500 ) ci1->rpm = 7500;
103:    if ( (ci1->rpm) < -7500 ) ci1->rpm = -7500;
104:
105:    ....
106: }
```

SX-WINDOW用ユーティリティ

どっち.X

Muroi Kouji 室井 幸治

SX-WINDOW使用時に，あるツールと相性の悪いソフトが起動していると，選択的に別のソフトを立ち上げるというツールです。これがあればシステムの再構築の手間がかなり軽減されます。

●

SX-WINDOW用にちょっと便利かもしれないプログラムを作成したので投稿してみました。

しばらく前のOh!Xの付録ディスクに，SXPICS.X，壁紙動画.R，壁動玉々.Rというプログラムがありましたね。これらは結構面白いプログラムなので，私もよく起動しています。

ただ，これらはプログラムの起動(終了)方法がちょっと面倒です。SXPICS.Xは「PICファイルの表示」ということで，目的はキャンバス.Xとほぼ同じです。しかし，アイコンメンテではPICファイルの起動コマンドにSXPICS.Xかキャンバス.Xのどちらか一方しか登録できません。また，壁紙動画.Rは終了するとき，わざわざコマンドラインに“-R”をつけなければなりません。これではちょっと不便ではないかと思って作成したプログラムが今回の「どっち.X」です。

使い方

このプログラムは，SX-WINDOW上で任意のタスクが起動されているかどうかをチェックし，その状況に応じた実行ファイルを起動するためのものです。アイコンメンテなどで起動ファイルの指定に使うと便利でしょう。リスト1をアセンブルするか，リスト2のダンプリストを打ち込んで(858バイト)，LHA.Xで展開すれば実行ファイルが現れます。

起動は，

どっち.X　タスク名　実行ファイル1　実行ファイル2

で行います。「タスク名」は検索したいタスク名（実行ファイル名と同じ）で，これが起動されていたなら「実行ファイル名1」で示されるコマンドを起動し，そうでなければ「実行ファイル2」で示されるコマンドを起動します。

このとき，タスク名，実行ファイル名の拡張子（.X，.R）は省略できません。また，実行ファイル名はアポストロフィ（'）で囲むことによりオプションなどの一連のコマンドラインを付加できます。また，アポストロフィで囲まれた文字列中にアポストロフィを記述する場合には，「&'」のように直前に「&」を置いてください。

そのほか，実行ファイル名1，2はハイフン（－）を記述することで省略可能です(実行ファイル名2は特に記述しなくてもかまいません)。両方を省略した場合は，どっち.Xはなにもせずに終了します。

使用例

いくつか例を挙げてみましょう。アイコンメンテで，起動ファイル名はすべて「どっち.X」，実行オプションに以下の指定をしてみてください。

1)　GRW.Xが起動しているときのみキャンバス.Xを立ち上げる場合

GRW.X　'キャンバス.X　%'

2)　GRW.Xが起動しているときはキャンバス.X，していないときはSXPICS.Xを立ち上げる場合

GWR.X　'キャンバス.X　%'　'SXPICS.X　-f%'

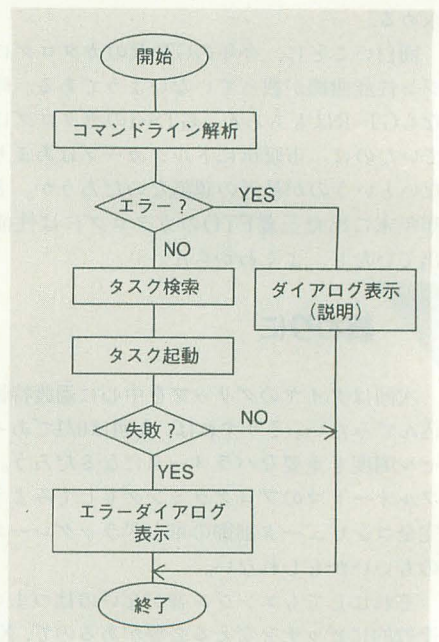
3)　GRW.Xが起動してないときのみ壁動玉々.Rを起動/終了する（トグル動作）

GRW.X　-　'どっち.X 壁紙動画.R　&'壁紙動画.R　-R&'　壁動玉々.R'

＊　　＊　　＊

ちなみに，1)，2)は＊.PICアイコン用，3)は壁動玉々.Rのアイコン用です。

プログラムの流れはフローチャートのとおりです。難しいことはやっていませんので，詳細はソースのほうを見てください。

図1 フローチャート

リスト1

```
 1: STACKSIZE  = 512
 2: SPACE                  = ' '
 3: SKIP                   = '-'
 4: TAB                    = $09
 5: APS                    = $27
 6: CHR                    = '&'
 7: DLG_X                  = 470
 8: DLG_Y                  = 200
 9: MES_X                  = 12
10: MES_Y                  = 16
11:
12:     .offset 0          *引数格納用
13: fileName:   ds.b    256             *ファイル名
14: swNum:              ds.b    2       *コマンドライン文字数
15: switch:             ds.b    256     *コマンドライン文字列
16: dialogHndl:         ds.l    1       *ダイアログ用(ハンドル)
17: dialogPtr: ds.l     1               *ダイアログ用(ポインタ)
18: dialogLoc: ds.w     4               *ダイアログ用
19: ARGSTR:
20:
21:     .offset 0       *ワークエリア
22: arg1:               ds.b    ARGSTR
23: arg2:               ds.b    ARGSTR
24: arg3:               ds.b    ARGSTR
25: errMes:             ds.b    256
26: WORKSIZE:
```

```
27:
28:     .text
29: header:
30:     dc.l    'OBJR'                  *ヘッダ
31:     dc.l    0                       *
32:     dc.l    main-header             *
33:     dc.l    WORKSIZE+STACKSIZE      *
34:     dc.l    0,0,0,0                 *
35:
36: start:
37:     pea.l   errmes(pc)              *Humanから起動されたとき
38:     dc.w    $ff09                   *
39:     addq.l  #4,sp                   *
40:     dc.w    $ff00                   *
41: errmes:  dc.b    'SX-SYSTEM上で起動して下さい',$0a,$0d,0
42:
43:     .even
44: main:
45:     movea.l a1,a5                   *ワークエリアのポインタ保存
46:
47:     movea.l a2,a0                   *コマンドライン解析
48:     bsr     getarg                  *
49:
50:     move.w  d0,d1                   *ステータスコピー
51:     cmpi.w  #2,d0                   *引数が1つ以下?
52:     bmi     error                   *そうならエラー
53:     swap    d0                      *
54:     tst.w   d0                      *引数に誤り?
55:     bne     error                   *そうならエラー
56:
57:     move.b  arg2(a5),d0             *実行ファイル名が
58:     beq     chkarg3                 *両方とも省略されているか
59:     subi.b  #SKIP,d0                *をチェック
60: chkarg3:                            *
61:     subi.w  #3,d1                   *3つ目の引数はある?
62:     beq     chkarg3_1               *あるときはchkarg3_1へ
63:     moveq.l #0,d1                   *
64:     bra     cmparg                  *
65: chkarg3_1:                          *
66:     move.b  arg3(a5),d1             *
67:     beq     cmparg                  *
68:     subi.b  #SKIP,d1                *
69: cmparg:                             *
70:     or.b    d1,d0                   *
71:     beq     error                   *省略されていたらエラー
72:
73:     cmpi.b  #SKIP,arg1(a5)          *省略文字?
74:     beq     error                   *そうなら終了
75:     cmpi.b  #0,arg1(a5)             *ヌル文字列
76:     beq     error                   *そうなら終了
77:
78:     move.w  #-1,-(sp)               *タスクを検索
79:     pea.l   arg1(a5)                *タスク名
80:     dc.w    $a3f4                   *タスクIDをD0.wに返す
81:     addq.l  #6,sp                   *
82:
83:     lea     arg2(a5),a0             *ファイル名のポインタ
84:     tst.w   d0                      *タスクは?
85:     bpl     exec                    *起動している
86:     lea     ARGSTR(a0),a0           *ファイル名のポインタを移動
87:
88: exec:
89:     cmpi.b  #SKIP,(a0)              *省略文字?
90:     beq     taskend                 *そうなら終了
91:     cmpi.b  #0,(a0)                 *ヌル文字列
92:     beq     taskend                 *そうなら終了
93:
94:     movea.l a0,a2                   *退避
95:     move.w  #0,-(sp)                *タスク起動
96:     move.l  #0,-(sp)                *
97:     move.l  #0,-(sp)                *
98:     pea.l   swNum+1(a0)             *コマンドライン
99:     move.l  a0,-(sp)                *ファイル名
100:    move.w  #0,-(sp)                *
101:    dc.w    $a351                   *
102:    lea     20(sp),sp               *
103:    tst.l   d0                      *
104:    bpl     taskend                 *正常終了のとき
105:
106:    lea     emsor(pc),a0            *起動できなかったとき
107:    lea     errMes(a5),a1           *
108:    move.b  (a0)+,(a1)+             *
109: errDlg_1:                          *
110:    move.b  (a2)+,(a1)+             *
111:    bne     errDlg_1                *
112:    subq.l  #1,a1                   *
113: errDlg_2:                          *
114:    move.b  (a0)+,(a1)+             *
115:    bne     errDlg_2                *
116:    pea.l   errMes(a5)              *
117:    move.w  #$0101,-(sp)            *
118:    dc.w    $a2f6                   *エラーダイアログ表示
119:    addq.l  #6,sp                   *
120:
121: taskend:
122:    move.w  #0,-(sp)                *終了コード
123:    dc.w    $a352                   *タスク終了
124:
125: error:
126:    bsr     dialog                  *ダイアログ表示
127:    bra     taskend                 *終了
128:
129: ***********************************************************
130: * d0        エラーコード
131: *           (d0.highはエラーコード。d0.lowは引数の数)
132: *
133:
134: getarg:
135:    movem.l d1-d2/a0-a1,-(sp)
136:
137:    move.b  #0,(a0)+                *+1(都合により文字数を0)
138:    lea     arg1(a5),a1             *コピー先アドレス
139:    moveq.l #0,d1                   *引数カウンタリセット
140:    moveq.l #3-1,d2                 *ループカウンタ
141:
142: getarg_1:
143:    bsr     gethead                 *頭だし
144:    cmpa.l  #0,a0                   *文字列終わり?
145:    beq     getarg_end              *終わりのとき
146:    addq.l  #1,d1                   *カウンタ+1
147:    bsr     brkcpy                  *コピー
148:    tst.l   d0                      *エラー?
149:    bne     getarg_end              *エラーのとき(d0.l = -1)
150:    lea     ARGSTR(a1),a1           *ポインタ移動
151:    dbra    d2,getarg_1             *ループ
152:
153: getarg_end:
154:    move.w  d1,d0                   *引数の数
155:    movem.l (sp)+,d1-d2/a0-a1
156:    rts
157:
158: ***********************************************************
159: * a0        単語の先頭アドレス
160: *           (移動する。値が0のときは文字列終わり)
161: *
162:
163: gethead_lop:
164:    addq.l  #1,a0                   *
165: gethead:
166:    cmpi.b  #SPACE,(a0)             *
167:    beq     gethead_lop             *ブランクのとき
168:    cmpi.b  #TAB,(a0)               *
169:    beq     gethead_lop             *タブのとき
170:    tst.b   (a0)+                   *
171:    bne     gethead_end             *頭だしOK!
172:    movea.l #1,a0                   *文字列終わり
173: gethead_end:
174:    subq.l  #1,a0                   *
175:    rts
176:
177: ***********************************************************
178: * d0.l      エラーコード(破壊する)
179: * a0        コピー元のポインタ(移動する)
180: * a1        コピー先のポインタ
181: *
182:
183: brkcpy:
184:    movem.l d1/a1-a3,-(sp)
185:    lea     swNum(a1),a2            *コマンドライン文字数
186:    lea     switch(a1),a3           *コマンドライン文字列
187:
188:    move.w  #0,(a2)                 *文字列数
189:    move.b  #0,(a3)                 *ヌル文字列
190:
191:    moveq.l #0,d0                   *ｱﾎﾟｽﾄﾛﾌｨ確認
192:    cmpi.b  #APS,(a0)               *
193:    bne     brkcpy_10               *
194:    moveq.l #-1,d0                  *ｱﾎﾟｽﾄﾛﾌｨあり
195:    addq.l  #1,a0                   *
196:    bsr     gethead                 *
197:    bra     brkcpy_11               *
198:
199: brkcpy_10:
200:    move.b  (a0)+,(a1)+             *先頭の単語(ﾌｧｲﾙ名)をコピー
201: brkcpy_11:
202:    cmpi.b  #SPACE,(a0)             *ブランク?
203:    beq     brkcpy_12               *
204:    cmpi.b  #TAB,(a0)               *タブ?
205:    beq     brkcpy_12               *
206:    cmpi.b  #APS,(a0)               *アポストロフィ?
207:    beq     brkcpy_12               *
208:    tst.b   (a0)                    *文字列終端?
209:    bne     brkcpy_10               *
210: brkcpy_12:
211:    move.b  #0,(a1)                 *文字列終端
212:    tst.l   d0                      *
213:    beq     brkcpy_end              *正常終了
214:
215: brkcpy_20:
216:    bsr     gethead                 *文字列(コマンドライン)の先頭
217:    cmpa.l  #0,a0                   *文字列の終端?
218:    bne     brkcpy_21               *
219:    moveq.l #-1,d0                  *エラーコード
220:    bra     brkcpy_end              *エラー終了
221:
222: brkcpy_21:
223:    moveq.l #0,d0                   *エラーコード
224:    moveq.l #0,d1                   *コマンドライン文字数
225: brkcpy_22:
226:    cmpi.b  #APS,(a0)               *ｱﾎﾟｽﾄﾛﾌｨ?
227:    bne     brkcpy_23               *違うとき
228:    cmpi.b  #CHR,-1(a0)             *一つ前が'文字記号'?
229:    bne     brkcpy_25               *違うなら終了
230:    subq.l  #1,a3                   *1文字戻す
231: brkcpy_23:
232:    tst.b   (a0)                    *文字列が途中で終了?
233:    beq     brkcpy_24               *そうならエラー
234:    move.b  (a0)+,(a3)+             *コピー
235:    addq.l  #1,d1                   *文字数カウント
236:    bra     brkcpy_22               *
237: brkcpy_24:
238:    moveq.l #-1,d0                  *エラーコード
239: brkcpy_25:
240:    addq.l  #1,a0                   *1文字進める
241:    move.w  d1,(a2)                 *カウント数保存
242:    move.b  #0,(a3)                 *終端文字
243:
244: brkcpy_end:
```

```
245:    movem.l (sp)+,d1/a1-a3
246:    rts
247:
248: ****************************************************************
249: *  引数・戻り値なし
250: *
251:
252: dialog:
253:    movem.l d0-d1/a0-a2,-(sp)
254:
255:    dc.w    $a35e                  *ダイアログの表示位置
256:    move.l  d0,dialogLoc(a5)       *(レクタングル)
257:    addi.l  #DLG_X*$10000+DLG_Y,d0 *
258:    move.l  d0,dialogLoc+4(a5)     *
259:
260:    move.l  #itemed-itemst,-(sp)   *メモリブロック確保
261:    dc.w    $a021                  *
262:    addq.l  #4,sp                  *
263:    move.l  d0,dialogHndl(a5)      *ハンドルを保存
264:    movea.l d0,a0                  *
265:    movea.l (a0),a0                *ポインタ取得
266:    lea     itemst(pc),a1          *アイテムリストをコピー
267:    move.l  #itemed-itemst-1,d0    *
268: dialog_1:                         *
269:    move.b  (a1)+,(a0)+            *
270:    dbra    d0,dialog_1            *
271:
272:    dc.w    $a360                  *ダイアログ表示
273:    move.l  dialogHndl(a5),-(sp)   *
274:    move.l  d0,-(sp)               *
275:    move.w  #0,-(sp)               *
276:    move.l  #-1,-(sp)              *
277:    move.w  #$26*16,-(sp)          *
278:    move.w  #-1,-(sp)              *
279:    pea.l   title(pc)              *
280:    pea.l   dialogLoc(a5)          *
281:    clr.l   -(sp)                  *
282:    dc.w    $a2c3                  *
283:    lea     30(sp),sp              *
284:    move.l  a0,dialogPtr(a5)       *
285:
286:    move.l  a0,-(sp)               *グラフポートセット
287:    dc.w    $a131                  *
288:    addq.l  #4,sp                  *
289:
290:    lea     messt(pc),a1           *ポインタのポインタ
291: dialog_2:
292:    movea.l (a1),a2                *メッセージのポインタ
293:    move.w  (a2),-(sp)             *FontFace
294:    dc.w    $a18c                  *
295:    move.l  2(a2),-(sp)            *文字列描画
296:    pea.l   6(a2)                  *
297:    dc.w    $a1a1                  *(影つき)
298:    lea     10(sp),sp              *
299:    addq.l  #4,a1                  *
300:    tst.l   (a1)                   *文字列終了?
301:    bne     dialog_2               *まだあるならループ
302:
303:    clr.l   -(sp)                  *「確認」が押されるまで待つ
304:    dc.w    $a2c7                  *
305:    addq.l  #4,sp                  *
306:
307:    move.l  dialogPtr(a5),-(sp)    *ダイアログ破棄
308:    dc.w    $a2c6                  *
309:    addq.l  #4,sp                  *
310:    move.l  dialogHndl(a5),-(sp)   *ヒープゾーン解放
311:    dc.w    $a038                  *
312:    addq.l  #4,sp                  *
313:
314:    movem.l (sp)+,d0-d1/a0-a2
315:    rts
316:
317: itemst:
318:    dc.w    1-1
319:    dc.l    0
320:    dc.w    DLG_X-43,DLG_Y-30,DLG_X-11,DLG_Y-11
321:    dc.b    4,6,4,'確認',0
322: itemed:
323: messt:
324:    dc.l    mes1,mes2,mes3,mes4,mes5,mes6,mes7,mes8,0
325: mes1:      dc.w    %0000,DLG_X/2-32,4
326:    dc.b    'どっち.X',0
327:    .even
328: mes2:      dc.w    %0000,MES_X,MES_Y*1+16
329:    dc.b    '機能',0
330:    .even
331: mes3:      dc.w    %0000,MES_X*2,MES_Y*2+16
332:    dc.b    '任意のタスクを検索し、それに応じた実行ファイルを起動
します。',0
333:    .even
334: mes4:      dc.w    %0000,MES_X,MES_Y*4+20
335:    dc.b    '使い方(コマンドライン)',0
336:    .even
337: mes5:      dc.w    %0100,MES_X*3,MES_Y*5+20
338:    dc.b    'どっち.x  検索タスク名  実行ファイル名1  実行ファイ
ル名2',0
339:    .even
340: mes6:      dc.w    %0000,MES_X*6,MES_Y*6+24
341:    dc.b    '検索タスク名: 検索するタスク名',0
342:    .even
343: mes7:      dc.w    %0000,MES_X*4,MES_Y*7+24
344:    dc.b    '実行ファイル名1: タスクが見つかったときに起動され
るファイル名',0
345:    .even
346: mes8:      dc.w    %0000,MES_X*4,MES_Y*8+24
347:    dc.b    '実行ファイル名2: タスクが見つからなかったときに起
動されるファイル名',0
348: title:     dc.b    8,'どっち.x',0
349: emsor:     dc.b    '「','」を起動できませんでした。',0
350:
351:    .end    start
```

リスト2

```
000000  21 A4 2D 6C 68 35 2D 36 : 5E
000008  03 00 00 A8 04 00 00 E5 : 94
000010  1E 37 1E 20 01 08 82 C7 : E5
000018  82 C1 82 BF 2E 78 DB 45 : 4A
000020  48 00 00 02 EE 6A A3 DA : 1F
000028  36 D4 87 EE FE 71 DA 58 : 20
000030  46 9C 70 B4 BE 24 E5 20 : ED
000038  B8 B4 ED 2A 98 A6 F0 40 : F1
000040  9D 2C 34 79 E2 33 69 25 : 19
000048  C1 DB 14 C4 80 9D D9 A9 : 13
000050  42 19 D4 29 4C 84 EF FB : 12
000058  C3 23 6E D0 4B 1C 3E 27 : F0
000060  49 20 6C 6D 42 86 29 75 : A8
000068  DB 6B 36 AA ED B4 CE 7C : 11
000070  4A 89 54 C1 50 42 DB 10 : 65
000078  A3 94 CA 26 60 77 DD D2 : AD
-------------------------------------
CKSUM:  B4 AB FB F5 B5 BD FA 7C 726E

000080  53 2A 1B 3E B4 6E 59 23 : 74
000088  4D CA A7 08 37 29 10 03 : 39
000090  56 0F 25 1E 1F FD 70 31 : 65
000098  15 95 77 52 80 17 98 7E : 20
0000A0  55 EA 65 83 D6 EF 16 A4 : A6
0000A8  D8 4B 59 29 7C 39 78 BB : 8D
0000B0  FF 6A 97 E2 5E DE 64 8C : 0E
0000B8  ED 02 E9 17 B5 47 AC 5D : F4
0000C0  12 D6 7D E7 06 07 B2 8A : 95
0000C8  B3 40 3B 1E B8 3F 4C 09 : 98
0000D0  EE 80 68 28 9A A9 AE 01 : F0
0000D8  CE 3D 22 20 C1 A6 80 24 : 58
0000E0  6C 40 78 C1 BC 70 98 F9 : A2
0000E8  04 8D 80 28 CC 04 A7 98 : 48
0000F0  C0 01 5F A4 09 00 31 94 : 92
0000F8  39 8D 31 3A 1C 26 DB 53 : A1
-------------------------------------
CKSUM:  0E 67 66 6F B5 27 86 4D 48A7

000100  20 02 BF 8B 1A 94 4D B2 : 19
000108  20 AA 78 E6 C4 DA 22 0F : F7
000110  CF 04 98 8E 38 83 11 A0 : 65
000118  29 0E 10 04 18 4E 11 B9 : 7B
000120  8A 9C 63 0C 82 F5 C4 B5 : 85
000128  E4 F0 3F 30 F3 D5 9F 8E : 38
000130  8C 4F 1B CE 24 5D B3 B0 : A8
000138  D0 D6 43 EF 5E DC BA 28 : F4
000140  6B 2F 6D 4C 83 B0 E1 C2 : 29
000148  30 CA DF BE 35 27 46 CA : 03
000150  18 E7 EA 99 E2 53 66 66 : 83
000158  A9 F6 C0 22 ED 80 71 C3 : 22
000160  90 49 BE 9F 4A DB B0 79 : 84
000168  F2 99 9B 84 AC F5 DE C5 : EE
000170  B4 44 24 FD 0D ED 61 19 : 8D
000178  1F 5D E1 D2 32 65 50 18 : 2E
-------------------------------------
CKSUM:  B3 C8 33 B3 E1 0E 9E 59 9D1E

000180  2D 16 3F 81 C6 F1 8F D5 : 1E
000188  5B F5 CE 47 92 24 0C 97 : BE
000190  42 46 4A 9B 29 CE DF DA : 1D
000198  30 3D 8B 46 13 7B A8 01 : 75
0001A0  7C 37 9C 02 27 D5 5D EC : 96
0001A8  E6 E5 50 9B DA 98 44 35 : A1
0001B0  9D 3E 7B A7 DF 38 AD 83 : 44
0001B8  55 3D 7D A4 04 5A B3 D8 : 9C
0001C0  C3 9B 52 E4 02 B9 B3 9B : 9D
0001C8  99 BB 9C 30 AB A7 DF 3F : 90
0001D0  DD 87 A8 DB 6B DE CB CC : C7
0001D8  54 F6 0E BE D6 94 CC CD : 19
0001E0  87 1A 7B D6 5C 72 C7 D7 : 5E
0001E8  BD 23 8F FD 5F C2 BF 17 : 63
0001F0  05 32 3E 3A 79 9F 68 59 : 88
0001F8  1D 37 AE F1 3D F5 4F 49 : BD
-------------------------------------
CKSUM:  41 9E 60 3C D7 F7 89 C6 1C3C

000200  6A 43 B4 88 A6 8F 06 2F : 53
000208  8C 39 28 B8 97 D0 3F EE : 39
000210  93 65 BB 75 E6 41 48 84 : 1B
000218  82 17 39 6D B1 A8 70 89 : 91
000220  31 DE 8F CB 21 CA 64 11 : C9
000228  F6 27 3E B7 CE 3C BB E8 : BF
000230  E0 A8 43 A0 BD AB EB 2D : EB
000238  48 58 72 20 49 86 85 5C : E2
000240  D8 38 00 45 31 C0 EA B5 : E5
000248  44 0B E5 A8 AB F0 AF B0 : D6
000250  23 FB AB 52 16 A9 D5 BB : 6A
000258  8E E6 4A 9B E5 A9 47 D7 : 05
000260  DD 3C 91 93 73 E6 33 33 : FC
000268  C3 A2 67 7C 2C 1A 71 BF : BE
000270  06 45 38 C2 84 9D 41 BD : 64
000278  91 BF D8 00 F2 FC 6F BE : 43
-------------------------------------
CKSUM:  5E 03 34 0F B5 1A 95 10 E05A

000280  1B E5 8D EA 0D F6 2E FD : A5
000288  35 DF 03 4B DD 2F CA 5E : 96
000290  CE F2 E7 DE AF C1 1C C4 : D5
000298  E9 E3 3B AE FC 30 A9 C5 : 4F
0002A0  A1 F5 97 E9 6B 8A D2 CD : AA
0002A8  48 2F 87 37 9B CA FD 8B : 22
0002B0  A4 CC 42 F5 CB F9 17 BD : 3F
0002B8  47 E4 2F 4C B6 39 7A BC : CB
0002C0  AF 79 AE 43 49 B5 15 AC : D8
0002C8  C2 F8 6E FD A5 E0 2E 9B : 73
0002D0  38 97 59 F8 BC B6 0B 5B : F8
0002D8  E2 FE 59 F7 1A 94 D7 BE : 73
0002E0  D6 73 5C 66 A8 88 BC E6 : DD
0002E8  71 C5 69 A1 00 3D BB AF : E7
0002F0  FC A6 53 71 BF 71 D7 CC : 39
0002F8  D4 65 3F ED B3 CC D4 2F : E7
-------------------------------------
CKSUM:  7D B6 66 B6 FA 7D 64 A5 1BC0

000300  07 CC 86 4D DF DC 72 BF : 92
000308  C2 F1 91 76 26 D3 2D AD : 8D
000310  C4 4E 06 C2 0A 1E 66 7C : E4
000318  64 6E CB 1D 1C DE 81 7E : B3
000320  62 F4 06 06 B0 5E E1 73 : C4
000328  CB DE 76 46 D1 19 92 D7 : B8
000330  90 85 C6 2F C7 E5 80 FF : 35
000338  D6 2E C9 6C FF CC B7 BB : 76
000340  93 A0 56 AF 77 8D 7B 62 : 19
000348  33 C0 5E A9 7F 41 D3 A4 : 31
000350  5B 0E 71 F9 29 8F 10 FF : 9A
000358  5E 00 00 00 00 00 00 00 : 5E
000360  00 00 00 00 00 00 00 00 : 00
000368  00 00 00 00 00 00 00 00 : 00
000370  00 00 00 00 00 00 00 00 : 00
000378  00 00 00 00 00 00 00 00 : 00
-------------------------------------
CKSUM:  03 6C 18 DA 91 30 8E 6F C161

総バイト数 = 858 バイト
```

WE WANT YOU!

Oh!Xは，読者の皆さん1人ひとりの力が作り上げていく雑誌です。あなたも誌面作りに協力してくれませんか？

投稿大募集

Oh!Xでは読者の皆さんによる投稿作品を常時募集しています。

未発表の作品であれば，グラフィック，音楽，システムプログラム，ツール，ゲーム，ハードウェアなどジャンルを問いません。機種についても特に限定はしませんが，雑誌の性格上扱いにくい場合もあります。

誌面に載りきらない大きなアプリケーションなどはディスクメディアを使って配布することが考えられます。その形態のひとつはご存じ付録ディスク，そしてもうひとつは別冊形式によるものです（発売中の「Z-MUSICシステムver.2.0」に続き，今後もいくつかのOh!X BOOKSシリーズが予定されています）。

また，「こんなものを作ってみました」といったものでもかまいません。気軽に作品を送ってみませんか。

投稿募集要項

1）お送りいただくプログラムには，住所，氏名，年齢，職業，連絡先電話番号，機種名，使用言語，動作に必要な周辺機器，パソコン歴などを明記のうえ，封書の宛先の最後には「Oh!X LIVE」「全機種共通システム」「投稿ゲームプログラム」など，プログラムの内容を明確にご記入ください。

2）投稿されるプログラムには詳しい内容を記入した原稿を同封してください。ディスクの中にドキュメントファイルの形式でのみ記述している方がいますが，郵送時の事故などでメディアが破壊されることもありますので，必ず文書を添えるようにしてください。変数表，メモリマップ，参考文献などの情報があればなお結構です。また，掲載に際しては，プログラムやデータ原稿に対して加筆修正をさせていただくことがあります。

3）お送りいただくプログラムは事故防止のため最低2回はセーブしておいてください。基本的に原稿などの返送はいたしませんので，あらかじめご了承ください。

4）ハード製作関係の投稿については，最初は内容のわかる原稿のみお送りいただければ結構です。その後，当方で製作物が必要だと判断した場合には改めてご連絡いたします。

5）作品の採用については，掲載号が決定した時点で当方より連絡いたします。特にツールやハード関係などの作品は特集内容などを考慮したうえで採用決定されますので，結果を連絡するまで時間がかかる場合があります。

6）投稿いただいたプログラムにバグなどが発見された場合は，新しいプログラムの入ったメディアと一緒に文書にてご連絡ください。

7）掲載されたプログラムに対しては当社規定の原稿料をお支払いいたします。また，投稿されたプログラムの著作権などはすべて制作者に保留されますが，いわゆる「フリーソフトなどとしてネットにアップする」ことなどを希望される場合には，必ず事前に編集部までご連絡ください。なお，一般的モラルとして，他誌との二重投稿，または他誌に掲載されたプログラムの移植などは固くお断りいたします。

その他，不明な点は編集部までお問い合わせください。

Oh!X編集部　☎03(5642)8122

協力スタッフ募集

Oh!Xでは誌面作りに参加していただく協力スタッフを募集しています。

スタッフとして活動する熱意があり，東京近郊にお住まいの方でソフトバンクに来社可能な方。時間的束縛は特にありませんが，ある程度時間に余裕がある方に限ります。基本的に学生を対象にしていますが，時間的余裕と余力が十分にあれば社会人も可とします。ただし，18歳未満の学生および浪人生の方については採用予定はありません。

応募要項ですが，ライター希望の方はOh!X誌面1ページ分相当（2500字程度）の自由論文に自己紹介文を添えて「Oh!Xスタッフ希望」係までお送りください。

また，文章力には自信がないけどプログラムなら……という方でも技術スタッフとして参加していただく場合があります。こちらを希望の方は，自由論文の代わりにこれまでに制作した自作プログラムとその解説などを一緒に応募してください。

書類選考後，採用の方にはこちらからご連絡いたします。

すべての読者へのお願い

いまはまだ何もできないけれど，いつかは……と思っているアナタにも，いますぐできるいちばん重要なことがあります。アンケートハガキへのご協力です。Oh!Xの誌面の方向性は，このアンケートで寄せられた読者のご意見をもとに決定されています。

皆さんからの熱いメッセージをお待ちしています。

そして，宛先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社
Oh!X編集部　○○○○係

イラスト投稿の規定

サイズはハガキ大（A6判）からB5判くらいまでを目安としますが，取り扱いの手間や現実的な問題としてハガキ大を一応の標準とします。いずれにせよ，掲載時にはかなり縮小されることを考慮して描いてください。

一応の推奨形式は以下のとおりです。

1）ハガキ大のケント紙で郵送

ハガキでも結構ですが，たまに裏面にも消し印が押される危険があります。

2）黒一色（薄ズミ不可）

墨汁は汚れの原因になることがあります。製図用インクがおすすめです。原稿は縮小されますのでスクリーントーンの80，90番台（レトラセットの場合）や色の濃すぎるものなどについての再現は保証しかねます。また，残念ながら，カラー原稿はごくたまにしか掲載されません。

内容に関して特に規制はありませんが，季節ものについては，掲載が予想される時期を考慮して早めに送ったほうが有利になることがあります（年賀状は例外）。

皆さんの力作をお待ちしております。

バックナンバー案内

BACK ISSUES

ここには1994年3月号から1995年2月号までをご紹介しました。現在1993年9，12月号，1994年1，4～12月号，1995年2月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は142ページを参照してください。

1994

3月号(品切れ)
特別企画　ひなまつりPRO-68K

連載
ハードコア3D/マシン語プログラミング/ゲーム作りのKNOW HOW
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/響子 in CGわ～るど/ファイル共有の実験と実践

●特別付録　ひなまつりPRO-68K（5″2HD）
●新製品紹介　ビデオPC for X680x0
LIVE in '94　THEME FROM WINNING RUN/スターフォースアレンジ版
THE SOFTOUCH　卒業/マッドストーカーX68/B-FIELD! 他
全機種共通システム　S-OSで学ぶZ80マシン語講座(4)

4月号
特集　SX-WINDOWの活用

連載
ハードコア3D/こちらシステムX探偵事務所
DōGA CGアニメーション講座/響子 in CGわ～るど
ショートプロ/ローテク工作/ANOTHER CG WORLD

●決定!　1993年度GAME OF THE YEAR
●新製品紹介　ビデオ入力ユニットCZ-6VS1
LIVE in '94　宇宙戦艦ヤマト/プロジェクトA子
THE SOFTOUCH　ジオグラフシール/ぷよぷよ/レッスルエンジェルス2　他
全機種共通システム　S-OSで学ぶZ80マシン語講座(5)

5月号
特別企画　こいのぼりPRO-68K
第9回言わせてくれなくちゃだワ

連載
ハードコア3D/響子 in CGわ～るど/ショートプロ
DōGA CGアニメーション講座/ファイル共有の実験と実践
こちらシステムX探偵事務所/ANOTHER CG WORLD

●特別付録　こいのぼりPRO-68K(5″2HD)
●新製品紹介　WorkroomSX-68K/開発キットツール集
LIVE in '94　ロード/時間旅行
THE SOFTOUCH　大魔界村/アルゴスの戦士/ジオグラフシール　他

6月号
特集　X68000と仲間たち

連載
ハードコア3D/響子 in CGわ～るど/ショートプロ
ローテク工作/ファイル共有の実験と実践
こちらシステムX探偵事務所/ANOTHER CG WORLD

●第5回Oh!Xアンケート分析大会
●新製品紹介　F-Calc for x68k
LIVE in '94　キャミイのテーマ/The End of Love
THE SOFTOUCH　スーパーリアル麻雀PIV/あすか120% BURNING Fest.他
全機種共通システム　YGCS ver.0.30

7月号
特集　入門コンピュータミュージック

連載
響子 in CGわ～るど/ショートプロ/ゲーム作りのKNOW HOW
ローテク工作/システムX探偵事務所/マシン語プログラミング
DōGA CGアニメーション講座/ファイル共有の実験と実践

●特別付録　CGA入門キット「GENIE」
●実用講座　Photo CDでカードを作る
LIVE in '94　宇宙刑事ギャバン/究極戦隊ダダンダーン/スティング　他
THE SOFTOUCH　麻雀航海記/雀神クエスト/The World of X68000 II 他
全機種共通システム　シューティングゲーム作成講座(1)

8月号
特集　Graphic Movement

連載
響子 in CGわ～るど/ショートプロ/ハードコア3D
ローテク工作/ANOTHER CG WORLD/善バビ
DōGA CGアニメーション講座/石の言葉，言葉の夢

●新製品紹介　X-SIMM VI/Mu-1 GS
SX-WINDOW ver.3.1
LIVE in '94　PURE GREEN/Ridge racer(POWER REMIX)
THE SOFTOUCH　Mr.Do!/Mr.Do! vs UNICORNS/レッスルエンジェルス3
全機種共通システム　シューティングゲーム作成講座(2)

9月号
特集　SX-WINDOW環境セットアップ

連載
響子 in CGわ～るど/ショートプロ/ハードコア3D
ローテク工作/DōGA CGアニメーション講座/善バビ
システムX探偵事務所/ファイル共有の実験と実践

●新製品紹介　X68030 D'ash/MJ-700V2C
●新刊紹介　X680x0 TeX
LIVE in '94　LOVE IS ALL/HELL HOUND/踏切の通過音
THE SOFTOUCH　餓狼伝説SPECIAL
全機種共通システム　怪しいZ80の使い方(テクニック編)

10月号
特別企画　もみじ狩りPRO-68K

連載
響子 in CGわ～るど/ショートプロ/ハードコア3D
TeX入門講座/ゲーム作りのKNOW HOW/善バビ
猫とコンピュータ/ファイル共有の実験と実践

●特別付録　もみじ狩りPRO-68K(5″2HD)
●新製品紹介　F-Card V5 for x68k
LIVE in '94　イース2/MSX用GRADIUS2/NATURE
THE SOFTOUCH　スーパーストII/スターラスター　他
全機種共通システム　怪しいZ80の使い方/ゲーム作成講座(3)

11月号
特集　STEP UP BASIC

連載
響子 in CGわ～るど/ショートプロ/ハードコア3D
TeX入門講座/DōGA CGアニメーション講座
システムX探偵事務所/ローテク工作/善バビ

●新製品紹介　BJC-400J/X680x0 Develop. & libc II
Free Software Selection Vol.2
LIVE in '94　ダーク・スペース/ENDLESS RAIN/レナのテーマ
THE SOFTOUCH　スーパーストII/餓狼伝説SPECIAL
全機種共通システム　B-GALETS2

12月号
特別企画　XL/Imageお試し版＋α

連載
響子 in CGわ～るど/ショートプロ/ハードコア3D
ファイル共有の実験と実践/DōGA CGアニメーション講座
システムX探偵事務所/ローテク工作/TeX入門講座

●特別付録　XL/Imageお試し版＋α(5″2HD)
●新製品紹介　H.A.R.P/XDTP SX-68K
LIVE in '94　幻想即興曲/きまぐれ オレンジ☆ロード 他
THE SOFTOUCH　魔法大作戦/スーパーストII
全機種共通システム　シューティングゲーム作成講座(4)

1995

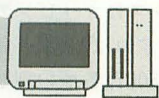
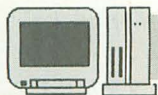
1月号(品切れ)
特集　割り切って使うCD-ROM

連載
響子 in CGわ～るど/ショートプロ/ハードコア3D
ファイル共有の実験と実践/DōGA CGアニメーション講座
システムX探偵事務所/ローテク工作/TeX入門講座

●CD-ROMドライブ紹介　CS-CD301X/CDS-E/SCD-200
●新製品紹介　X68000XVI用アクセラレータXellent30
LIVE in '95　ぷよぷよ/ジムノペディNO.1/PRIME
THE SOFTOUCH　パックランド/上海 万里の長城/魔法大作戦
餓狼伝説SP 特別編/スーパーストII 特別編

2月号
特集　MicroProcessingUnit

連載
響子 in CGわ～るど/ショートプロ/ハードコア3D
SX-BASIC公開デバッグ/DōGA CGアニメーション講座
システムX探偵事務所/SX-WINDOWによるDTP

●特別企画　最新ゲーム機を見る
●新製品紹介　Datacalc SX-68K/シャーペンワープロパック
●1994年度GAME OF THE YEARノミネート作品発表
LIVE in '95　サムライスピリッツ/AFTER SCHOOL/白鳥の湖
THE SOFTOUCH　スーパーストII 特別編

仮想ドライバの開発実験PART8.

絶対転送速度76,800bpsへの挑戦

電机本舗 由井 清人 Yui Kiyoto

今回はSCC直接駆動によるポーリング処理をドライバに組み込み，76,800bpsでの高速通信をめざします。また，いままで訂正を重ねてきた仮想ドライブシステムの全ソースリストを掲載します。

今回は，前回確立した高速化技術を具体的に仮想ドライバに組み込む実験を行います。

まず，結論から先にいいましょう。従来IOCSコールを使っていたRS-232C制御ルーチンを，SCC直接駆動ルーチンにしただけで，ほぼ問題なく動作しました。それも，転送速度76,800bpsでです。

当初，SCC直接駆動によるポーリング処理ですと，LSIのもっている通信バッファが2バイト（事実上は1バイトと考えたほうが無難でしょう）と小さいため，文字の取りこぼしがあるのではないか，と不安でした。従来のIOCSを利用した通信ですと，システムが十分大きな通信バッファを用意してくれるので文字あふれの不安がなかったわけです。

しかし，実際にはそれまで使用していたIOCS関数のLOF232C(),INP232C(),OUT232C()の通信制御関数を自作のSCC制御関数_LOF232C(), _INP232C(), _OUT232C()に変更しただけでだいたい動作しました。

実行前は，非常に不安でしたので，まず手始めに従機側の制御ソフト「R.C」だけを改造してみました。通信速度は，とりあえず従来と互換性を確保するために9,600bpsに設定して行いました。結果は良好でするする動きます。このときに主機側には前回作った仮想ドライブシステムを組み込んでテストを行いました。

これに気をよくして（緊張をいくらかといて）主機側の仮想ドライブプログラムに同様の処理をしました。改良の詳細は後述しますが，結果は良好で同じくきれいに動いてくれます。おまけに驚いたことに動作が非常に安定しました。これまで，よく原因不明のシステムエラーが出ることがあったのですが，これがほとんど出なくなったのです。

いままで，CHKDSKを仮想ドライブにかけると多発していたシステムエラーが嘘のように解消されました。本来ならば，IOCSは通信バッファが効いているので文字あふれがないはずなのですが，割り込みによる受信が災いしてか文字の取りこぼしが起きていたようです。おそらくは割り込み処理のオーバーヘッドが大きく，動作が不安定になっていたのでしょう。

ここで初めて，通信速度の高速化実験を行ってみました。最初は，主従双方の機械を無難に36,800bpsに設定してみました。結果は良好です。もしやと思い76,800bpsにしてみたところ，ばっちり動きました。さらに，DISKCOPYを使い，仮想ドライブ間のデータ転送をかけてみたところ，2HDでジャスト5分という数字が出ました。これより逆算して実効転送速度を出してみましょう。

2HDは，1,024バイトのセクタが1,221個より構成されています。ですから，ディスク1枚の容量は，1,250,304バイトとなります。かかった時間は5分ということから300秒。したがって次のようになります。

実効速度：41,588.7＝(1,250,304/300)×10

FD間の転送で約4万bps出ていることがわかります。ただし，FDの読み取りで1分，書き込み処理でさらに1分くらい，データ通信と異なるところで時間がかかっていることを考慮してください。そうすると，実際の転送を3分であると考えるとだいたい7万bps出ていることになります。これは，ほぼ理論どおりの速度を実現できているといえるでしょう。

この転送速度ですと，ほとんどRS-232Cを意識しないで使えるようです。もちろん，仮想ドライブからのプログラム起動や，ファイル転送が遅いのはいまだに変わりません。ですが，頻繁に使用するディレクトリ一覧やカレントドライブやディレクトリ移動は，ほとんど通常のドライブと変わらないようです。

ただし，残念なことに，今回の高速化が仇になりSX-WINDOW上では使用できなくなりました。詳しくは追求していませんが，仮想ドライブを組み込んでSX-WINDOWを起動すると，マウスが一切効かなくなります。

MFPの説明

次に，前回約束したとおり，MFPの説明をしましょう。MFPは，マルチファンクションペリフェラルの略で，モトローラの68901というLSIをX68000で採用しています。このMFPには，CPUの割り込み制御とタイマほか，あると便利な機能が入っておりCPUの動作を助けています（図1）。

このMFPは，本編とは本来は関係のない部品ないし機能です。しかし，高速通信をするうえでは考慮しないわけにはいかない理由があります。

まず，仮想ドライブシステムが76,800bpsで高速動作しているときに，割り込みが発生した場合を考えてみてください。CPUは割り込みが発生すると現在行っている処理（たとえば仮想ドライブとのデータ交換）を中断し

てほかのことを始めます。そして，ほかの処理が終わったならば，元の処理を開始するという具合です。

たいていの普通の処理は中断されても問題になりません。あとでいくらでも再開可能だからです。しかし例外もあります。通信がその典型といえるでしょう。もしもデータを連続受信しているときに，割り込みが発生したらどうなるのでしょうか。まず，CPUが連続受信を中断しても，送信側にはそれがわかりません。ですから送信側はかまわずデータを流し続けます。CPUの割り込み処理が終わり，受信を再開したときにはあらかたデータはあふれ落ちていることになります。このようなわけで，データ通信，それもSCCを直接駆動するような場合には，割り込みは天敵となるわけです。

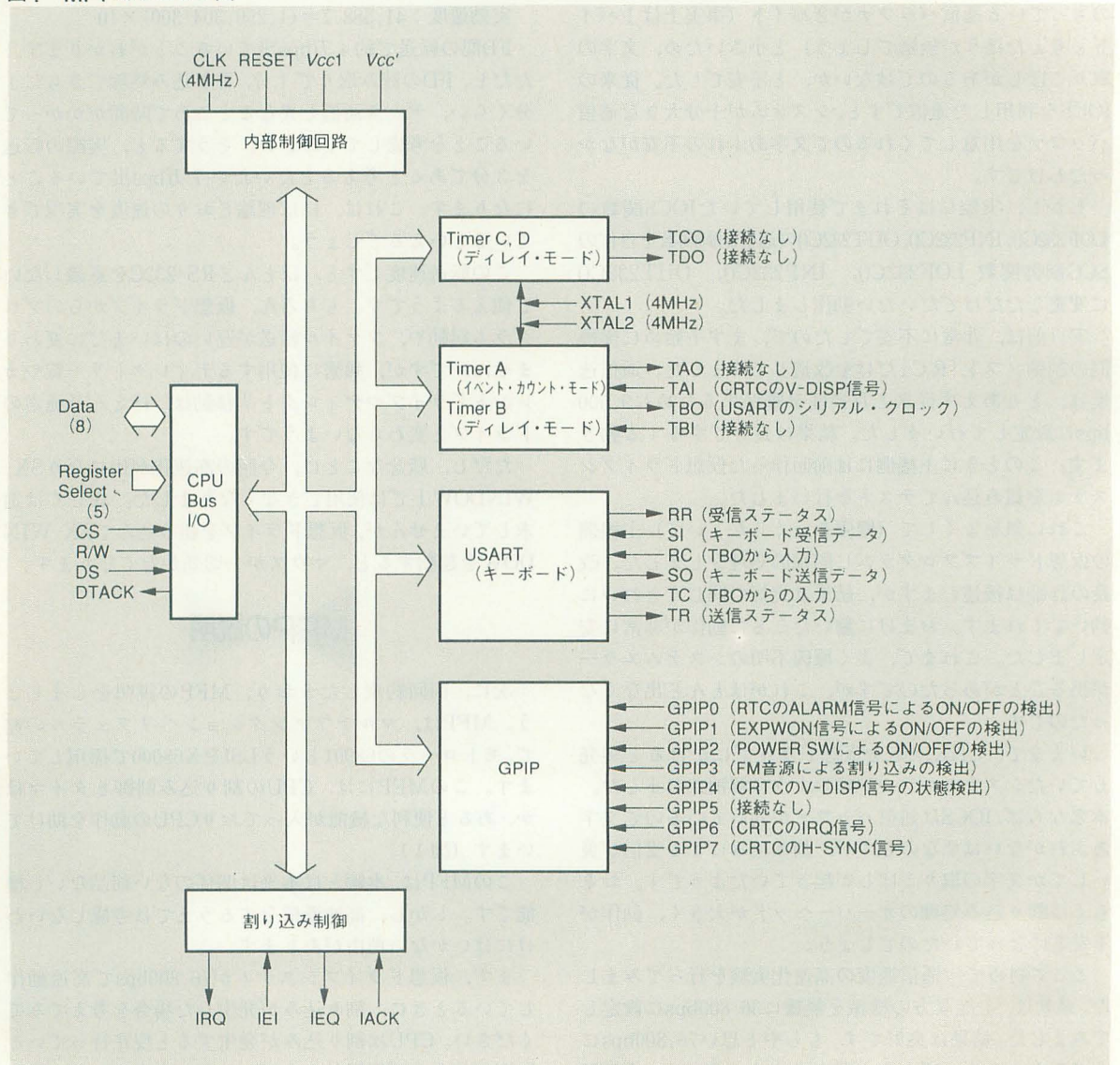
MFPは先ほど述べたように，割り込みの制御をします。ですから，通信をしていて割り込みの制御をしたいときには，MFPの設定を変更する必要があるわけです。

図2にMFPの割り込み関係を示します。これより割り込みのレベルにより0～15まで合計16の割り込みの存在がわかります。できることなら，通信の間は，これらすべてを冬眠させてしまいたいものです（精神衛生上割り込みを止めておくのが好ましい）。

図3にMFPを具体的に制御するレジスタを示します。それぞれは，メインメモリのアドレスにマッピングされていて，該当するメモリを読み書きすればMFPと命令のやりとりを行えます。この中で，E88007$_H$番地の割り込みイネーブルレジスタAとE88009$_H$番地の割り込みイネーブルレジスタBを注目してください。ここで，各割り込みの許可（イネーブル），不許可（ディセーブル）のコントロールを行っているのです。

次にこの2つのレジスタの扱い方を図4に示します。

通信するにあたりいちばん安全なのは，割り込みイネーブルレジスタA，Bの両方にゼロを書き込んでしまうことです。そうすれば，すべての割り込みは禁止されます。しかし，全部の割り込みを禁止するのは気が引けるので，CPUに対する負荷の軽そうなキーボード処理あたりは残しておくことにします。ここでリスト4の1133行からの関数xdisable()を見てください。ここで，割り込みイネーブルレジスタAには，0x1e（00011110$_B$）を設定

図1　MFPのブロック図

し，キーボード関係を残して割り込みを禁止しています。そして割り込みイネーブルレジスタBにはゼロを書き込み，すべての割り込みを禁止しています。

この関数は，仮想ドライブが通信を行うときに発行し，割り込みから通信データの保護を行っています。ちなみに，リスト4の1138行からの関数_rcv1()はその逆です。事前に，int_a_sts，int_b_stsへ読み込んでおいたMFPの設定を，2つのレジスタに書き込んでいます。この関数は，データ通信が終わったときに発行し，割り込みの冬眠を解除する役割をしています。

図2　MFPの割り込み系ブロック図

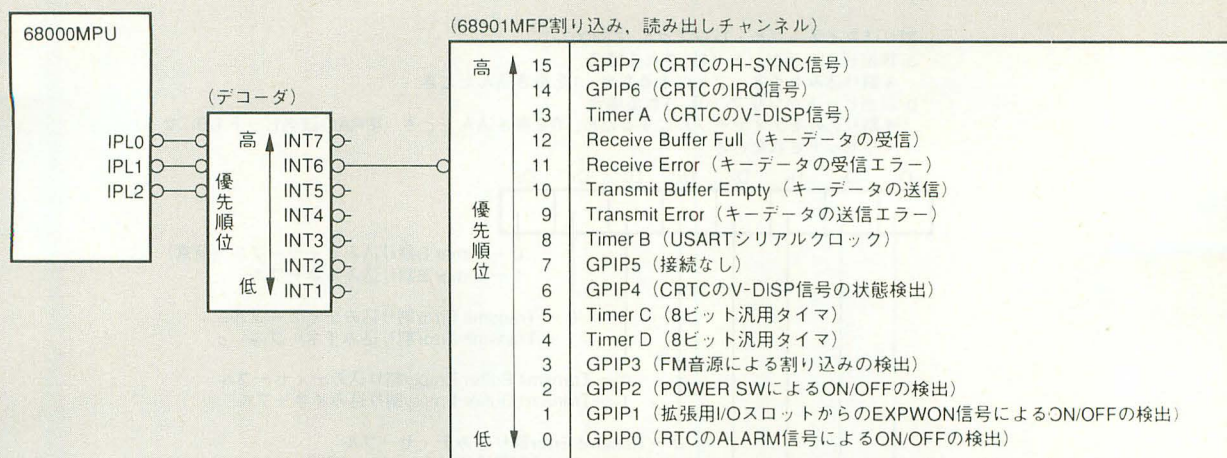



図3　MFPレジスタアドレスマップ

	レジスタ・アドレス	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	備考
GPIPコントロール	E88001$_H$	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0	GPIPデータ・レジスタ（Read only）
	E88003$_H$	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0	アクティブ・エッジ・レジスタ（AER）
	E88005$_H$	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0	データ・ディレクション・レジスタ（DDR）
割り込みコントロール	E88007$_H$	GPIP7	GPIP6	TimerA	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TimerB	割り込みイネーブル・レジスタ（IERA）
	E88009$_H$	GPIP5	GPIP4	TimerC	TimerD	GPIP3	GPIP2	GPIP1	GPIP0	割り込みイネーブル・レジスタ（IERB）
	E8800B$_H$	GPIP7	GPIP6	TimerA	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TimerB	割り込みペンディング・レジスタA（IPRA）
	E8800D$_H$	GPIP5	GPIP4	TimerC	TimerD	GPIP3	GPIP2	GPIP1	GPIP0	割り込みペンディング・レジスタB（IPRB）
	E8800F$_H$	GPIP7	GPIP6	TimerA	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TimerB	割り込みインサービス・レジスタA（ISRA）
	E88011$_H$	GPIP5	GPIP4	TimerC	TimerD	GPIP3	GPIP2	GPIP1	GPIP0	割り込みインサービス・レジスタB（ISRB）
	E88013$_H$	GPIP7	GPIP6	TimerA	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TimerB	割り込みマスク・レジスタA（IMRA）
	E88015$_H$	GPIP5	GPIP4	TimerC	TimerD	GPIP3	GPIP2	GPIP1	GPIP0	割り込みマスク・レジスタB（IMRB）
	E88017$_H$	V_7	V_6	V_5	V_4	S				ベクタ・レジスタ
タイマコントロール	E88019$_H$				Reset TAO	AC3	AC2	AC1	AC0	タイマAコントロール・レジスタ（TACR）
	E8801B$_H$				Reset TBO	BC3	BC2	BC1	BC0	タイマBコントロール・レジスタ（TBCR）
	E8801F$_H$		CC2	CC1	CC0		DC2	DC1	DC0	タイマCコントロール・レジスタ（TCDCR）
	E8801F$_H$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	タイマAコントロール・レジスタ（TADR）
	E88021$_H$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	タイマBコントロール・レジスタ（TBDR）
	E88023$_H$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	タイマCコントロール・レジスタ（TCDR）
	E88025$_H$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	タイマDコントロール・レジスタ（TDDR）
USARTコントロール	E88027$_H$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	同期キャラクタ・レジスタ（未使用）
	E88029$_H$	CLK	WL1	WL0	ST1	ST0	PE	E/O	＊	USARTコントロール・レジスタ（UCR）
	E8802B$_H$	BF	CE	PE	FE	F/SorB	M/CIP	SS	RE	受信ステータス・レジスタ（RSR）
	E8802D$_H$	BE	UE	AT	END	B	H	L	TE	送信ステータス・レジスタ（TSR）
	E8802F$_H$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	USARTデータ・レジスタ（UDR）

プログラムのコンパイル

今回は，部分的にあちこち修正しましたので，主従すべてのプログラムをリストアップします。これらのプログラムのコンパイルは次のように行います。

cc /Y /Ns D0.S D1.C D3.C D2.S

従機側のコンパイルは次のように行ってください。

cc /Y /Ns R.C D3.C >tmp.tmp

D3.Cは主機側で使っているものと同じです。

図4 MFPレジスタの内容

・割り込みイネーブル・レジスタA（E88007H）
a. 該当ビットがセット（1）される場合
＊割り込みをイネーブルにするため，1を書き込んだとき
b. 該当ビットがクリア（0）される場合
＊割り込みをディセーブルにするため，0を書き込んだとき（IPRAの該当ビットも0になる）
＊リセットされたとき

D7 D6 D5 D4 D3 D2 D1 D0

0 ― Timer B割り込みディセーブル（通常）
1 ― Timer B割り込みイネーブル

0 ― Transmit Error割り込みディセーブル
1 ― Transmit Error割り込みイネーブル

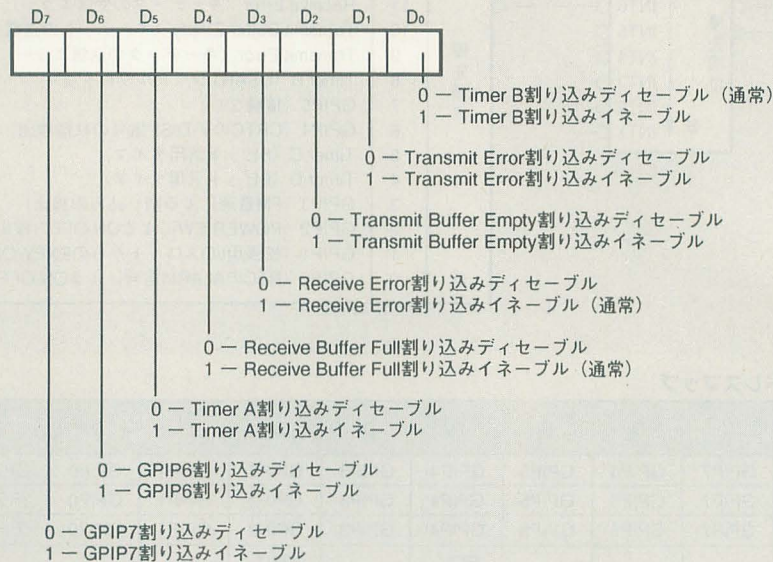
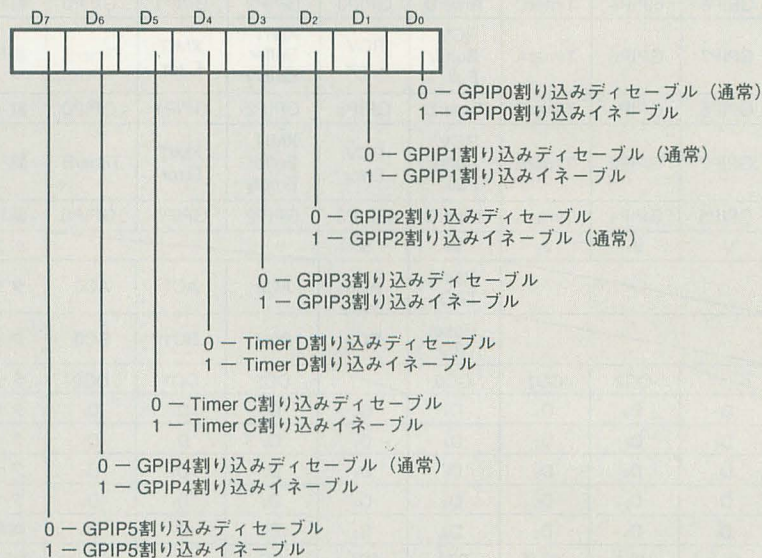
0 ― Transmit Buffer Empty割り込みディセーブル
1 ― Transmit Buffer Empty割り込みイネーブル

0 ― Receive Error割り込みディセーブル
1 ― Receive Error割り込みイネーブル（通常）

0 ― Receive Buffer Full割り込みディセーブル
1 ― Receive Buffer Full割り込みイネーブル（通常）

0 ― Timer A割り込みディセーブル
1 ― Timer A割り込みイネーブル

0 ― GPIP6割り込みディセーブル
1 ― GPIP6割り込みイネーブル

0 ― GPIP7割り込みディセーブル
1 ― GPIP7割り込みイネーブル

・割り込みイネーブル・レジスタB（E88009H）
a. 該当ビットがセット（1）される場合
＊割り込みをイネーブルにするため，1を書き込んだとき
b. 該当ビットがクリア（0）される場合
＊割り込みをディセーブルにするため，0を書き込んだとき（IPRAの該当ビットも0になる）
＊リセットされたとき

プログラムの説明

これまで開発してきたものを，くどくど述べても意味がないでしょう。ここでは，今回変更を加えた部分だけを解説します。

今回より，通信速度が向上したのは最初に述べたとおりです。これをもう少し詳しく説明しましょう。まず仮想ドライブシステムは，9,600bpsで立ち上がります。そして，主機と従機は連結を行います。次に，従機は自分のとりうる最高の転送速度を通知します。そしてその速度が9,600bpsより高速であれば，SCCの設定を変更し速度をアップします。ここではX68000同士の接続ですから，9,600bpsから76,800bpsにスピードアップしているわけです。

リスト1を参照してください。このソースD0.Sはデバイスドライバの本体をなすアセンブラプログラムです。この中で52行の部分は，主機側の仮想ドライブの初期化を行っているところです。従来，ここでは通信速度を9,600bpsに設定する初期化を行ってきました。ですが，今回は少し事情が異なります。開発中気がついたのですが，ここは起動直後の1回だけ呼ばれるはずなのに，どうも仮想ドライブをアクセスしている最中に頻繁に呼ばれているのです。

少し考えてみてください。たとえば，うまく主機と従機が連結し76,800bpsに速度アップしたときに，もしもここを再び実行したならば，主機側は通信速度を強制的に9,600bpsにダウンすることになります。しかし，従機側の通信速度は76,800bpsのままです。転送速度が異なると当然通信は不可能となるので，接続は切れてしまいます。ですから，ここが2回目以降に呼ばれないようにしないといけません。結局，変数をもたせ，初めに呼ばれたときだけ，通信の初期化を行い，2度目以降はなにもしないでスキップするようにしました。これが，48行から50行目です。

```
48： move.w  _inz_flg,d0   ＊RS-232C初期化flg
49： cmp.w   #0,d0
50： bne     dskstr_end    ＊RS-232C初期化
```

見てのとおり変数として，_inz_flgをもたせ初回呼び出しか否かを判定しています。もし2回目以降であれば，通信の初期化処理をスキップして終了するように変更しました。また，65行目を見てください。ここで，Cで記述したSCC初期化関数_rs_inz()を呼び，9,600bpsのポーリングモードにSCCを設定しています。

リスト2のD1.Cは，主機側の本体とも呼べる部分です。Cで記述しており，実際の処理はこのソースで行っています。574行から699行にかけて今回新しい関数を追加しました。主にSCCの制御に関する関数です。ここにある前回作成した関数_rs_inz()，_INP232C()，_OUT232C()は説明するまでもないでしょう。それぞれ，初期化，1文字受信，1文字送信を行います。

_rs_spd()は今回新しく作った関数です。これはSCCの速度の変更を行います。

_rcv()は設定を一度変更したSCCを元に戻す機能があります。

_LOF232Cは受信バッファにデータの有無をチェックします。オリジナルのIOCSのLOF232C()関数のSCC直接制御版です。ただし，SCC上の受信バッファ（1バイトバッファとして考える）のデータの有無を調べるので返す値は0か1です。このあたりは互換性維持で問題となるところです。

リスト4の従機側プログラムR.Cの変更は，主に512行から始まる初期化関数dskini()です。ここは，従機のもっているディスク装置を識別し仮想ドライブに登録する役割をしています。

536行から542行を見てください。ここはX68000上でコンパイルしたときに有効になる行です。機能は，537行で自分のとりうる最高の転送速度を主機へ通知しています。そして，538行でいくらか遅延時間を入れて，539行で自分（従機）の通信速度を76,800bpsに変更しています。ここで当然のことながら，ケーブルの向こう側でも同じく通信速度を76,800bpsに変更しています。

そして，次行で画面に速度アップのメッセージを出し再び遅延を入れています。遅延を入れているのは，CPUの速度に比べ，SCCの応答が十分遅いことが考えられるからです。

また，再び537行を見てください。ここで，1文字送信をしていますが，9,600bpsでデータを送信するわけですから，SCCは約1ms(1/1000秒)かけてデータを送ります。少なくともCPUはその1000倍は速いですから，もし，538行に遅延がなければ，539行の通信速度の変更は，データの送信中に行われることになります。このようなとき，SCCがどのような動作をするか不明です。したがって遅延を入れ，送信が終わったであろうタイミングを計っているわけです。

543行から548行はPC-9801上でコンパイルしたとき有効になるものです。ここでは，X68000のときと同様の処理をしますが，従来どおり9,600bpsに上限を設定しています。このプログラムはPC-9801でも動作を確認しています。

600行から625行を見てください。ここはPC-9801用に作ったものです。ここでは，DOSのバージョンを見て，DPBテーブルの作成を使い分けています。前回の調査で判明したとおりMS-DOSは，バージョン3までと，バージョン4以上でDPBの取り方に互換性がありません。ここでは，バージョンをチェックし，正しいDPBを得るようにしています。

946行にある関数getver()でバージョンを取得し，バージョン3以下であれば，602行から612行までを実行。そうでなければバージョン4なので，614行から624行を実行するわけです。

＊　　＊　　＊

以上で，仮想ドライブシステムが完成しました。来月は，完成した仮想ドライブシステムの具体的な動作試験と，SX-WINDOW上で動作をさせるための改良を行ってみます。

参考文献

「X68000ベストプログラミング」，千葉憲明著，技術評論社

リスト1　D0.S

```
  1: ******************************************************************
  2: *       DEVICE_DRIVER SAMPLE No1
  3: *       XCプログラマーズマニュアル.P643のSRAMDISKプログラム
  4: *       を改造して作りました。
  5: ******************************************************************
  6:         .include        doscall.mac
  7:         .include        iocscall.mac
  8:         .text
  9:
 10: .xref   _dskent
 11: .xref   _dskini
 12: .xref   _mediac
 13: .xref   _dskctrl
 14: .xref   _dskout
 15: .xref   _dskotv
 16: .xref   _dskinp
 17: .xref   _notcom
 18: .xref   __rs_inz
 19:
 20: .xdef   _d_dat
 21: .xdef   _d_tim
 22: .xdef   _d_dte
 23: .xdef   _inittbl
 24: .xdef   _rdmaxr
 25:
 26: dsktbl: dc.l    -1
 27:         dc.w    $0000
 28:         dc.l    dskstr
 29:         dc.l    dskent
 30:         dc.b    1,'S_RAM123'
 31: dskreq: dc.l    0
 32: dskjmp: dc.l    _dskini
 33:         dc.l    _mediac
 34:         dc.l    _notcom
 35:         dc.l    _notcom
 36:         dc.l    _dskinp
 37:         dc.l    _dskctrl
 38:         dc.l    _notcom
 39:         dc.l    _notcom
 40:         dc.l    _dskout
 41:         dc.l    _dskotv
 42:         dc.l    _notcom
 43:         dc.l    _notcom
 44:         dc.l    _notcom
 45:
 46: dskstr: move.l  a5,dskreq
 47:         movem.l d0-d7/a0-a6,-(sp) * 各レジスタをバックアップ
 48:         move.w  _inz_flg,d0       * rs232c初期化flg
 49:         cmp.w   #0,d0
 50:         bne     dskstr_end        * --- rs232c 初期化 ---
 51:         moveq.l #_SET232C,d0      * IOCS コール番号セット
 52:         move.w  #%0100_1100_0000_0111,d1 * stop 1
 53:                                          * parity none
 54:                                          * bit 8
 55:                                          * xon/xoff none
 56:                                          * bps 9600
 57:         trap    #15
 58:                                   * --- 'X'をテストで232cへ出力 ---
 59: *       moveq.l #_OUT232C,d0      * IOCS コール番号セット
 60: *       move.b  #$58,d1           * 'X'をテスト送出
 61: *       trap    #15
 62:
 63:         move.l  #3,d0             * C関数へ引数を渡す.3=9600bps set.
 64:         move.l  d0,-(sp)
 65:         jsr     __rs_inz          * C関数をコール
 66:         addq.l  #4,SP   * 後始末(引数受け渡しで使ったスタックを処分)
 67:
 68:         move.w  #1,d0
 69:         move.w  d0,_inz_flg
 70:
 71: dskstr_end:
 72:         movem.l (sp)+,d0-d7/a0-a6 * 各レジスタを復旧
 73:         rts
 74:
 75: dskent:
 76:         movem.l d0-d7/a0-a6,-(sp) * 各レジスタをバックアップ
 77:
 78:         move.l  dskreq,a5         * C関数へ引数を渡す
 79:         move.l  a5,-(sp)
 80:         jsr     _dskent           * C関数をコール
 81:         addq.l  #4,SP   * 後始末(引数受け渡しで使ったスタックを処分)
 82:
 83:         movem.l (sp)+,d0-d7/a0-a6 * 各レジスタを復旧
 84:         rts                       * Humanへ帰る.さやうならぁ.
 85:
 86: _inz_flg:       dc.w    0
 87:
 88: bpbtbl1:
 89:         dc.w    0
 90:         dc.b    0
 91:         dc.b    0
 92:         dc.w    0
 93:         dc.w    0
 94: _rdmaxr:        dc.w    0
 95:         dc.b    0
 96:         dc.b    0
 97:
 98: bpbtbl2:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
 99: bpbtbl3:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
100: bpbtbl4:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
101: bpbtbl5:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
102: bpbtbl6:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
103: bpbtbl7:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
104: bpbtbl8:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
105: bpbtbl9:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
106: bpbtbl10:       dc.b    0,0,0,0,0,0,0,0,0,0,0,0
107:
108: _inittbl:
109:         dc.l    bpbtbl1
110:         dc.l    bpbtbl2
111:         dc.l    bpbtbl3
112:         dc.l    bpbtbl4
113:         dc.l    bpbtbl5
114:         dc.l    bpbtbl6
115:         dc.l    bpbtbl7
116:         dc.l    bpbtbl8
117:         dc.l    bpbtbl9
118:         dc.l    bpbtbl10
119:
120: _d_dat: dc.b    'SRAM_DISK  ',$08,0,0,0,0
121:         dc.b    0,0,0,0,0,0
122: _d_tim: dc.b    $A0,$60
123: _d_dte: dc.b    $E7,$0A
124:         dc.b    0,0,0,0,0,0
125:
126:         end
```

リスト2　D1.C

```
  1: #define DEBUG   1
  2:
  3: #define BPS1200         0
  4: #define BPS2400         1
  5: #define BPS4800         2
  6: #define BPS9600         3
  7: #define BPS19200        4
  8: #define BPS38400        5
  9: #define BPS76800        6
 10:
 11: #include        <doslib.h>
 12: #include        <stdio.h>
 13: #include        <time.h>
 14:
 15: extern  short   d_tim;
 16: extern  short   d_dte;
 17: extern  char    d_dat[];
 18:
 19: /*
 20: extern  struct  BPBPOI  *inittbl;
 21: */
 22:
 23: extern  char    inittbl;
 24: extern  char    inittblend;
 25: extern  unsigned short  rdmaxr;
 26:
 27: #define SRAM    0xed0000
 28: #define SRAMMD  0xed002d
 29: #define SYS00d  0xe8e00d
 30:
 31: #define TIMEOUT 5L
 32:
 33: #define REQLEN  0
 34: #define UNITCD  1
 35: #define COMCOD  2
 36: #define ERRLOW  3
 37: #define ERRHIGH 4
 38:
 39: #define MXUNIT  13
 40: #define DEVEND  14
 41: #define BPBPOI  18
 42: #define BDEVNO  22
 43:
 44: #define DISKID  13
 45: #define DISKFG  14
 46:
 47: #define DMAADR  14
 48: #define DMALEN  18
 49: #define STAREC  24
 50: #define GETDAT  13
 51:
 52: char    *_devmes = "C: $ED0400-15K";
 53: char    *_mesini = ")初期化しました ¥n¥r";
 54: char    *_mesnoi = ")初期化しませんでした ¥n¥r";
 55:
 56: int     dskinp();
 57: int     dskout();
 58: int     dskini();
 59: int     mediac();
 60: int     dskpaw0();
 61: int     dskpaw1();
 62: int     dskpaw2();
 63: int     dskotv();
 64: long    _abs();
 65: void _rs_buf_clr();
 66: int notcom();
 67:
 68: int qout232c( int c );
 69:
 70: int     _rs_inz();
 71: void _rcv();
 72: int     _LOF232C();
 73: int     _INP232C();
 74: void    _OUT232C();
 75: void xenable();
 76: void xdisable();
 77: void _rcv1();
 78:
 79: /*----- d3.c func -----*/
 80: int     blk_in();
 81: int     blk_in1();
 82: int     blk_out();
 83: int     blk_out1();
 84: void    _rs_buf_clr();
 85:
 86: struct REQ_HED {
 87:         char    reqlen;
 88:         char    unitcd;
 89:         char    comcod;
 90:         char    errlow;
 91:         char    errhigh;
 92: };
 93:
 94: struct REQ_INI {
 95:    char   reqlen;   /* リクエストヘッダの長さ */
 96:    char   unitcd;   /* ユニットコード */
 97:    char   comcod;   /* コマンドコード */
 98:    char   errlow;   /* エラーコードその1 */
 99:    char   errhigh;  /* エラーコードその2 */
100:    char   rsv[8];   /* 予約領域 */
101:                     /* ---------------------------- */
102:    char   mxunit;   /* ユニット数 */
103:    char   *devend;  /* デバイスドライバ制御プログラムの終了アドレス */
104:    char   *bpbpoi;  /*struct  BPBPOI *bpbpoi;   BPBテーブルアドレス */
105:    char   bdevno;   /* ブロックデバイス番号(0=A*) */
106: };
107:
108: struct REQ_CHG {
109:    char   reqlen;   /* リクエストヘッダの長さ */
110:    char   unitcd;   /* ユニットコード */
111:    char   comcod;   /* コマンドコード */
112:    char   errlow;   /* エラーコードその1 */
113:    char   errhigh;  /* エラーコードその2 */
114:    char   rsv[8];   /* 予約領域 */
115:                     /* ---------------------------- */
116:    char   diskid;   /* よくわかんない<現在未使用> */
117:    long   diskfg;   /* よくわかんない 資料ではバイト型 */
118: };
119:
120: struct REQ_RW {
121:    char   reqlen;   /* リクエストヘッダの長さ */
122:    char   unitcd;   /* ユニットコード */
123:    char   comcod;   /* コマンドコード */
124:    char   errlow;   /* エラーコードその1 */
125:    char   errhigh;  /* エラーコードその2 */
126:    char   rsv[8];   /* 予約領域 */
127:                     /* ---------------------------- */
128:    char   diskid;   /* よくわかんない<現在未使用> */
129:    char   *dmaadr;  /* 転送バッファ */
130:    long   dmalen;   /* 転送セクター数 */
131:    char   starec00[2]; /* アクセスセクタ番号上位2byte */
132:    long   starec;      /* アクセスセクタ番号 */
133: };
134:
135: struct REQ_CTRL {
136:    char   reqlen;          /* リクエストヘッダの長さ */
137:    char   unitcd;          /* ユニットコード */
138:    char   comcod;          /* コマンドコード */
139:    char   errlow;          /* エラーコードその1 */
140:    char   errhigh;         /* エラーコードその2 */
141:    char   rsv[8];          /* 予約領域 */
142:                            /* ------------------------- */
143:    char   getdat;
144: };
145:
146: struct REQ_RW req_hed;
147:
148: #if DEBUG
149:
150: char    buf[2048];
151: /****************************************************************
152:    main() テスト用のメインルーチン
153:    エントリー(entry)は全プログラムの入り口と解釈すること
154:    ASMを強引にハンドディスコンパイル
155: ****************************************************************/
156: void main()
157: {
158:   int    sts;
159:   int    s;
160:    for( s=0 ; s<10 ; s++ ) {
161:       req_hed.reqlen = sizeof(struct REQ_CHG); /* リクエストヘッダの長さ *

162:       req_hed.comcod = s;
163:       switch( s ) {
164:          case 0:
165:             sts = dskini( &req_hed );
166:             break;
167:          case 1:
168:             sts = mediac( &req_hed );
169:             break;
```


▶冬休みに学校にもち込んでいたX68000を家にもち帰ったのだが，ディスプレイの後ろの隙間からゴキブリが……ハッとして目が覚めた。夢でよかった。でもディスプレイを分解して掃除するのは難しそうだし，どうしよう。　　西尾　昌人(20)愛知県

```
170:          case 4:
171:             req_hed.reqlen = sizeof(struct REQ_RW);
172:             req_hed.dmaadr = buf;
173:             req_hed.starec00[0] = req_hed.starec00[1] = 0;
174:             req_hed.dmalen = 1L;
175:             req_hed.starec = 1L;
176:             sts = dskinp( &req_hed );
177:             break;
178:          case 5:
179:             req_hed.reqlen = sizeof(struct REQ_CTRL);
180:             sts = dskctrl( &req_hed );
181:             break;
182:          case 8:
183:             req_hed.reqlen = sizeof(struct REQ_RW);
184:             buf[0] = 9;
185:             req_hed.dmaadr = buf;
186:             req_hed.starec00[0] = req_hed.starec00[1] = 0;
187:             req_hed.dmalen = 17L;
188:             req_hed.starec = 1L;
189:             sts = dskout( &req_hed );
190:             break;
191:          case 9:
192:             req_hed.reqlen = sizeof(struct REQ_RW);
193:             sts = dskotv( &req_hed );
194:             break;
195:          default:
196:             sts = notcom(&req_hed);            /* 未使用命令 */
197:             break;
198:       }
199:    }
200: }
201: #endif
202:
203: /*************************************************************
204:     diskent   世に言うエントリールーチン（舶来語1）
205:     エントリー(entry)は全プログラムの入り口と解釈すること
206:     ASMを強引にハンドディスコンパイル
207: *************************************************************/
208: void dskent( req_hed )
209: struct REQ_HED *req_hed;
210: {
211:    int   sts;
212:     xdisable();
213:     switch( req_hed->comcod ) {
214:        case 0:
215:           sts = dskini( req_hed );
216:           break;
217:        case 1:
218:           sts = mediac( req_hed );
219:           break;
220:        case 2:
221:           sts = notcom(req_hed);            /* 未使用命令         */
222:           break;
223:        case 3:
224:           sts = notcom(req_hed);            /* 未使用命令         */
225:           break;
226:        case 4:
227:           sts = dskinp( req_hed );
228:           break;
229:        case 5:
230:           sts = dskctrl( req_hed );
231:           break;
232:        case 6:
233:           sts = notcom(req_hed);            /* 未使用命令         */
234:           break;
235:        case 7:
236:           sts = notcom(req_hed);            /* 未使用命令         */
237:           break;
238:        case 8:
239:           sts = dskout( req_hed );
240:           break;
241:        case 9:
242:           sts = dskotv( req_hed );
243:           break;
244:        case 10:
245:           sts = notcom(req_hed);            /* 未使用命令         */
246:           break;
247:        case 11:
248:           sts = notcom(req_hed);            /* 未使用命令         */
249:           break;
250:        case 12:
251:           sts = notcom(req_hed);            /* 未使用命令         */
252:           break;
253:        default:
254:           sts = notcom(req_hed);            /* 未使用命令         */
255:           break;
256:     }
257:     xenable();
258: }
259:
260: /****************************************
261:     notcom    未使用命令
262:     ASMを強引にハンドディスコンパイル
263: ****************************************/
264: int notcom(req)
265: struct REQ_HED *req;
266: {
267:     req->errlow = 0x03;     /* 下位バイトエラーコード格納 */
268:     req->errhigh = 0x50;    /* 上位バイトエラーコード格納 */
269:     return( 0 );
270: }
271:
272: /*********************************************************
273:     mediac    メディア交換処理(RAMなので未使用のはず)
274: *********************************************************/
275: int mediac( req_chg )
276: struct REQ_CHG *req_chg;
277: {
278:    int   sts;
279:    int   n;
280:    char  c;
281:     req_chg->diskfg = 1L;              /* エラーコードセット */
282:     n = (int)(req_chg->reqlen);
283:     if( qout232c( 'S' ) ) {            /* 送信開始コード送出 */
284:        ;
285:     }
286:     else if( (sts=blk_in( &c, 1) )) {  /* 応答コード受信 */
287:        ;
288:     }
289:     else if( c != 'A' ) {
290:        ;
291:     }
292:     else if( (sts=blk_out( &n, sizeof(n) ))) { /* リクエストヘッダ長送信 */
293:        ;
294:     }
295:     else if( (sts=blk_out( req_chg, req_chg->reqlen ))) { /* リクエスト
ヘッダ送信 */
296:        ;
297:     }
298:     else if( (sts=blk_in( &(req_chg->diskfg), sizeof(req_chg->diskfg
) ))) {
299:        req_chg->diskfg = 1L;                    /* エラーコードセット   */
300:     }
301:     return( 0 );
302: }
303:
304: /****************************************
305:     dskinp    ディスク入力処理(?)
306:     ASMを強引にハンドディスコンパイル
307: ****************************************/
308: int dskinp( req )
309: struct REQ_RW *req;
310: {
311:    int   sts;
312:    int   n;
313:    int   len;
314:    int   wk;
315:    char   *p;
316:    char   c;
317:     req->errlow = 0x03;             /* 下位バイトエラーコード格納 */
318:     req->errhigh = 0x50;            /* 上位バイトエラーコード格納 */
319:     n = (int)(req->reqlen);
320:     if( qout232c( 'S' ) ) {         /* 送信開始コード送出 */
321:        ;
322:     }
323:     else if( (sts=blk_in( &c, 1) )) {  /* 応答コード受信 */
324:        ;
325:     }
326:     else if( c != 'A' ) {
327:        ;
328:     }
329:     else if( (sts=blk_out( &n, sizeof(n) ))) { /* リクエストヘッダ長送信 */
330:        ;
331:     }
332:     else if( (sts=blk_out( req, req->reqlen ))) { /* リクエストヘッダ送信
*/
333:        ;
334:     }
335:     else if( (sts=blk_in( &len, 4 ))) {
336:        ;
337:     }
338:     else {
339:        p = req->dmaadr;
340:        while( len>0 ) {
341:           wk = len;
342:           if( wk>16384L ) {
343:              wk = 16384L;
344:           }
345:           if( (sts=blk_in( p, wk ))) {
346:              break;
347:           }
348:           p += 16384;
349:           len -= 16384;
350:        }
351:        if( len<=0 ) {
352:           req->errlow = 0;     /* 下位バイトエラーコード格納 */
353:           req->errhigh = 0;    /* 上位バイトエラーコード格納 */
354:           sts = 0;
355:        }
356:     }
357:     return( sts );
358: }
359:
360: /****************************************
361:     dskotv    出力&ベリファイ処理(?)
362:     ASMを強引にハンドディスコンパイル
363: ****************************************/
364: int dskotv( req )
365: struct REQ_RW *req;
366: {
367:     return( dskout( req ) );
368: }
369:
370: /****************************************
371:     dskout    ディスク出力処理(?)
372:     ASMを強引にハンドディスコンパイル
373: ****************************************/
374: int dskout( req )
375: struct REQ_RW *req;
376: {
377:    int   sts;
378:    int   n;
379:    int   len;
380:    int   wk;
381:    char   *p;
382:    char   c;
383:    short   int   *b_no;
384:    char   **wk1;
385:    char   *bpb;
386:     req->errlow = 0x03;          /* 下位バイトエラーコード格納 */
387:     req->errhigh = 0x50;         /* 上位バイトエラーコード格納 */
388:     n = (int)(req->reqlen);
389:     if( qout232c( 'S' ) ) {   /* 送信開始コード送出 */
390:        ;
391:     }
392:     else if( (sts=blk_in( &c, 1) )) { /* 応答コード受信 */
393:        ;
394:     }
395:     else if( c != 'A' ) {
396:        ;
397:     }
398:     else if( (sts=blk_out( &n, sizeof(n) ))) { /* リクエストヘッダ長送信 */
399:        ;
400:     }
401:     else if( (sts=blk_out( req, req->reqlen ))) { /* リクエストヘッダ送信
*/
402:        ;
403:     }
404:     else {
405:        wk1 = (char**)(&inittbl);
406:        bpb = *wk1;
407:        bpb += (req->unitcd * 12);
408:        b_no = (short int*)bpb;
409:        len = req->dmalen;          /* len = 転送バイト数算出 */
410:        len *= (*b_no);             /* len = dmalen * 1024 */
411:        if( (sts=blk_out( &len, 4 ))) {    /* データ長送信 */
412:           ;
413:        }
414:        else {
415:           p = req->dmaadr;
416:
417:           while( len>0 ) {
418:              wk = len;
419:              if( wk>16384L ) {
420:                 wk = 16384L;
421:              }
422:              if( (sts=blk_out( p, wk ))) {    /* データ長送信 */
423:                 break;
424:              }
425:              c=_INP232C();
426:              p += 16384;
427:              len -= 16384;
428:           }
429:           if( len<=0 ) {
430:              if( (sts=blk_in( &req->errlow, 2 ))) { /* エラーコードセット
*/
431:                 ;
432:              }
433:              else if( req->errlow || req->errhigh ) {
434:                 ;
435:              }
436:              else {
437:                 req->errlow = 0;     /* 下位バイトエラーコード格納 */
438:                 req->errhigh = 0;    /* 上位バイトエラーコード格納 */
439:                 sts = 0;
440:              }
441:           }
442:        }
443:     }
444:     return( sts );
445: }
446:
447: /*************************************
448:     dskctrl    ディスクコントロール
449: *************************************/
450: int dskctrl( req )
451: struct REQ_CTRL *req;
452: {
453:    int   sts;
454:    int   n;
455:    int   len;
456:    char   c;
457:     req->errlow = 0x03;          /* 下位バイトエラーコード格納 */
458:     req->errhigh = 0x50;         /* 上位バイトエラーコード格納 */
459:     n = (int)(req->reqlen);
460:     if( qout232c( 'S' ) ) {   /* 送信開始コード送出 */
461:        ;
462:     }
463:     else if( (sts=blk_in( &c, 1) )) { /* 応答コード受信 */
464:        ;
465:     }
466:     else if( c != 'A' ) {
467:        ;
468:     }
469:     else if( (sts=blk_out( &n, sizeof(n) ))) {  /* リクエストヘッダ長送信 *
/
470:        ;
471:     }
472:     else if( (sts=blk_out( req, req->reqlen ))) { /* リクエストヘッダ送信
*/
473:        ;
474:     }
475:     else if( (sts=blk_in( req, req->reqlen ))) { /* リクエストヘッダ受信 *
/
476:        ;
477:     }
```

▶「XL/Image」はすばらしい。しかし，買ったはいいが，まだ使っていない。かまたさんに申しわけない。だれかXLモデラを作ってくれ〜(他力本願モード)。

竹内　孝雄(32)大阪府

```
478:     else {
479:         sts = 0;
480:     }
481:     return( sts );
482: }
483:
484: /************************************
485:     dskini    ディスク初期化ルーチン
486: ************************************/
487: int dskini( req )
488: struct REQ_INI *req;
489: {
490:    int    sts;
491:    int    n;
492:    char    u_no;
493:    char    drv_flg;
494:    char    **wk;
495:    char    *bpb;
496:    char    c;
497:    int    spd;
498:     _rs_buf_clr();                  /* rs232c buf clr */
499:     n = (int)(req->reqlen);
500:     req->errlow = 0x03;             /* 下位バイトエラーコード格納 */
501:     req->errhigh = 0x50;            /* 上位バイトエラーコード格納 */
502:     u_no = 0;
503:     wk = (char**)(&inittbl);
504:     bpb = *wk;
505:     if( qout232c( 'S' ) ) {    /* 送信開始コード送出 */
506:         ;
507:     }
508:     else if( (sts=blk_in( &c, 1) )) { /* 応答コード受信 */
509:         ;
510:     }
511:     else if( c != 'A' ) {
512:         ;
513:     }
514:     else if( (sts=blk_outl( &n, sizeof(n) ))) {  /* リクエストヘッダ長送信
*/
515:         ;
516:     }
517:     else if( (sts=blk_out( req, req->reqlen ))) { /* リクエストヘッダ送信
*/
518:         ;
519:     }
520:     else {
521:         PRINT( (unsigned char*)"¥n¥r仮想ドライブシステム登録開始¥n¥r" );
522:         spd = _INP232C();
523:         if( spd!=BPS9600 ) {
524:             _rs_spd( spd );
525:         }
526:         while( 1 ) {
527:             if( (sts=blk_in( &drv_flg, 1 ))) { /* bps status flg受信 */
528:                 break;
529:             }
530:             else if( drv_flg == -1 ) { /* normal end */
531:                 if( u_no==0 ) {
532:                     break;
533:                 }
534:                 req->errlow = 0;           /* 下位バイトエラーコード格納 */
535:                 req->errhigh = 0;          /* 上位バイトエラーコード格納 */
536:                 req->mxunit = u_no;        /* 最大unit セット */
537:                 req->bpbpoi = &inittbl;     /* BPBテーブルアドレスセット */
538:                 req->devend = &inittblend; /* デバイステーブルエンドアドレスセット
*/
539:                 break;
540:             }
541:             else {
542:                 if( (sts=blk_in( bpb, 12 ))) { /* bpb table受信 */
543:                     break;
544:                 }
545:                 bpb += (char*)(12);
546:                 u_no ++;
547:                 PRINT( (unsigned char*)("  仮想ドライブを1基追加 ¥n¥r") );
548:             }
549:         }
550:     }
551:     return( 0 );
552: }
553: int qout232c( int c )
554: {
555:    int    j;
556:    int    i;
557:    int    sts;
558:     sts = -1;
559:     _OUT232C( c );
560:     for( j=0 ; j<3 ; j++ ) {
561:         for( i=0 ; i<7000 ; i++ ) {
562:             if( _LOF232C() ) {
563:                 sts = 0;
564:                 break;
565:             }
566:         }
567:         if( sts==0 ) {
568:             break;
569:         }
570:         _OUT232C( c );
571:     }
572:     return( sts );
573: }
574: /* ======== 1994.12.13 修正 =========*/
575: int    __mode;
576: int    _spd_tbl[] = { 128, 63, 31, 14, 6, 2, 0 };
577: BYTE   int_a_sts;
578: BYTE   int_b_sts;
579: /************************
580:    _rs_inz        初期化
581:    return    :    sts
582: ************************/
583: _rs_inz( spd )
584: int    spd;        /* speed    */
585: {
586:     __mode = SET232C( -1 );
587:     SET232C( 0x4c07 );
588:     B_BPOKE(0xe98005, 0x03);    /* 受信モードセット */
589:     B_BPOKE(0xe98005, 0xe1);
590:     B_BPOKE(0xe98005, 0x05);    /* 送信モードセット */
591:     B_BPOKE(0xe98005, 0xea);
592:     B_BPOKE(0xe98005, 0x0c);    /* ボーレートセット */
593:     B_BPOKE(0xe98005, _spd_tbl[spd] );
594:     B_BPOKE(0xe98005, 0x0d);
595:     B_BPOKE(0xe98005, 0x00);
596:     B_BPOKE(0xe98005, 0x01);    /* ポーリングモードセット */
597:     B_BPOKE(0xe98005, 0x01);
598:     int_a_sts = B_BPEEK( 0xe88007 );
599:     int_b_sts = B_BPEEK( 0xe88009 );
600:     return( 0 );
601: }
602:
603: /**************************
604:    _rs_spd        速度変更
605:    return    :    sts
606: **************************/
607: _rs_spd( spd )
608: int    spd;        /* speed    */
609: {
610:     B_BPOKE(0xe98005, 0x03);    /* 受信モードセット */
611:     B_BPOKE(0xe98005, 0xe1);
612:     B_BPOKE(0xe98005, 0x05);    /* 送信モードセット */
613:     B_BPOKE(0xe98005, 0xea);
614:     B_BPOKE(0xe98005, 0x0c);    /* ボーレートセット */
615:     B_BPOKE(0xe98005, _spd_tbl[spd] );
616:     B_BPOKE(0xe98005, 0x0d);
617:     B_BPOKE(0xe98005, 0x00);
618:     B_BPOKE(0xe98005, 0x01);    /* ポーリングモードセット */
619:     B_BPOKE(0xe98005, 0x01);
620:     return( 0 );
621: }
622:
623: /***********************************
624:    _rcv()    通信割り込み設定の復旧
625:    return    : char, or sts
626: ***********************************/
627: void _rcv()
628: {
629:     B_BPOKE(0xe98005, 0x01);
630:     B_BPOKE(0xe98005, 0x11);
631:     B_BPOKE( 0xe88007, int_a_sts );
632:     B_BPOKE( 0xe88009, int_b_sts );
633:     SET232C( __mode );
634: }
635:
636: /*******************************************
637:     _LOF232C    受信バッファチェックルーチン
638: *******************************************/
639: int    _LOF232C()
640: {
641:   long    i;
642:   int    sts;
643:    sts = 0;
644:    B_BPOKE(0xe98005, 0);
645:    if( B_BPEEK(0xe98005)&0x01 ) {
646:        sts = 1;
647:    }
648:    return( sts );
649: }
650:
651: /*********************************
652:     _INP232C    1文字受信ルーチン
653: *********************************/
654: int    _INP232C()
655: {
656:   long    i;
657:   int    sts;
658:    sts = -1;
659:    for( i=0 ; i<1000000L ; i++ ) {
660:        B_BPOKE(0xe98005, 0);
661:        if( B_BPEEK(0xe98005)&0x01 ) {
662:            sts = B_BPEEK(0xe98007);
663:            break;
664:        }
665:    }
666:    return( sts );
667: }
668:
669: /**********************************
670:     _OUT232C    1文字送信ルーチン
671: **********************************/
672: void    _OUT232C( int c )
673: {
674:   long    i;
675:   int    sts;
676:    sts = -1;
677:    for( i=0 ; i<1000000L ; i++ ) {
678:        B_BPOKE(0xe98005, 0);
679:        if( B_BPEEK(0xe98005)&0x04 ) {
680:            B_BPOKE(0xe98007, (unsigned char)c); /* 1文字送出 */
681:            sts = 0;
682:            break;
683:        }
684:    }
685: }
686: void xenable()
687: {
688:     _rcvl();
689: }
690: void xdisable()
691: {
692:     B_BPOKE( 0xe88007, 0x1e );
693:     B_BPOKE( 0xe88009, 0x00 );
694: }
695: void _rcvl()
696: {
697:     B_BPOKE( 0xe88007, int_a_sts );
698:     B_BPOKE( 0xe88009, int_b_sts );
699: }
```

リスト3 D2.S

```
1:
2:          .text
3:          .data
4: .xdef    _inittblend
5:
6: _inittblend:
7:          dc.b    0
8:
9:          end
```

リスト4 R.C

```
 1: #define          X68      0
 2: #define          PC98     1
 3: #define          IBM      2
 4: #define          SYSTEM   0
 5:
 6: #define DEBUG    1
 7: #define ESC      0x1b
 8:
 9: #define BPS1200          0
10: #define BPS2400          1
11: #define BPS4800          2
12: #define BPS9600          3
13: #define BPS19200         4
14: #define BPS38400         5
15: #define BPS76800         6
16:
17: #if SYSTEM==X68
18: #include          <doslib.h>
19: #include          <stdio.h>
20: #include          <time.h>
21: #define           blk_in_x(a,b)   blk_in(a,b)
22: #define           blk_out_x(a,b)   blk_out(a,b)
23: #define XCHG2(a)
24: #define XCHG4(a)
25: void DISKREDX( unsigned char* buf, int drv, long rec, long len );
26: void DISKWRTX( unsigned char* buf, int drv, long rec, long len );
27: #endif
28:
29: #if SYSTEM==PC98 || SYSTEM==IBM
30: /*#pragma pack(1)*/
31: #include          <stdlib.h>
32: #include          <dos.h>
33: #include          <sys¥types.h>
34: #include          <sys¥stat.h>
35:
36: typedef long                INT;
37: typedef unsigned long       UINT;
38: typedef short               WORD;
39: typedef unsigned short      UWORD;
40: typedef char                BYTE;
41: typedef unsigned char       UBYTE;
42: typedef void                VOID;
43: typedef int                 DEFAULT;
44: typedef int                 BOOLEAN;
45: typedef long                LONG;
46: typedef unsigned long       ULONG;
47:
48: struct  DPBPTR  {                    /* DOS Ver2.xx or 3.x */
49:         UBYTE   drive;
50:         UBYTE   unit;
51:         UWORD   byte;
52:         UBYTE   sec;
53:         UBYTE   shift;
54:         UWORD   fatsec;
55:         UBYTE   fatcount;
56:         UWORD   dircount;
57:         UWORD   datasec;
58:         UWORD   maxfat;
```

▶２月号の特集はなかなかよいと思います。しかし，こういうのばかりだと困るのですが。
山内　斉(24)宮城県

```
 59:            UBYTE   fatlen;
 60:            UWORD   dirsec;
 61:            INT     driver;
 62:            UBYTE   id;
 63:            UBYTE   flg;
 64:
 65:            char    next[4];          /*struct DPBPTR *next;*/
 66:            UWORD   dirfat;
 67:            UBYTE   dirbuf[64];
 68: };
 69:
 70: struct  DPBPTR1 {                    /* DOS Ver 4.XX later   */
 71:            UBYTE   drive;
 72:            UBYTE   unit;
 73:            UWORD   byte;
 74:            UBYTE   sec;
 75:            UBYTE   shift;
 76:            UWORD   fatsec;
 77:            UBYTE   fatcount;
 78:            UWORD   dircount;
 79:            UWORD   datasec;
 80:            UWORD   maxfat;
 81:            UWORD   fatlen;
 82:            UWORD   dirsec;
 83:            INT     driver;
 84:            UBYTE   id;
 85:            UBYTE   flg;
 86:
 87:            char    next[4];          /*struct DPBPTR *next;*/
 88:            UWORD   dirfat;
 89:            UBYTE   dirbuf[64];
 90: };
 91:
 92: int        GETDPB( int drv, struct DPBPTR  *d );
 93: void DISKREDX( unsigned char* _rw_buf, int drv, long rec, long len
);
 94: int  diskred( unsigned char* _rw_buf, int drv, long rec, long len )
;
 95: void DISKWRTX( unsigned char* _rw_buf, int drv, long rec, long len
);
 96: int  diskwrt( unsigned char* _rw_buf, int drv, long rec, long len )
;
 97: long DRVCTRL( long mode, int drv );
 98: int        blk_in_x();
 99: int        blk_out_x();
100: void xchg2( char *data );
101: void xchg4( char *data );
102: long CURDRV();
103: long CHGDRV(long drv);
104:
105: #define XCHG2(a)          xchg2((char*)(a))
106: #define XCHG4(a)          xchg4((char*)(a))
107: void getver( int *ver, int *sub ver );
108:
109: #endif
110:
111: int         _rs_inz();
112: void _rcv();
113: int         _LOF232C();
114: int         _INP232C();
115: void        _OUT232C();
116: void xenable();
117: void xdisable();
118: void _rcv1();
119: void dlytime();
120:
121: #define TIMEOUT 5L
122:
123: #define REQLEN  0
124: #define UNITCD  1
125: #define COMCOD  2
126: #define ERRLOW  3
127: #define ERRHIGH 4
128:
129: #define MXUNIT  13
130: #define DEVEND  14
131: #define BPBPOI  18
132: #define BDEVNO  22
133:
134: #define DISKID  13
135: #define DISKFG  14
136:
137: #define DMAADR  14
138: #define DMALEN  18
139: #define STAREC  24
140: #define GETDAT  13
141:
142: void       rcv_ent();
143: int        r_dskinp();
144: int        r_dskout();
145: int        r_dskini();
146: int        r_mediac();
147: int        r_dskpaw0();
148: int        r_dskpaw1();
149: int        r_dskpaw2();
150: int        r_dskotv();
151:
152: long       _abs();
153: void _rs_buf_clr();
154: int notcom();
155:
156: /*----- d3.c funo -----*/
157: int        blk_in();
158: int        blk_in1();
159: int        blk_out();
160: int        blk_out1();
161: void       _rs_buf_clr();
162:
163: struct REQ_HED {
164:            char    reqlen;
165:            char    unitcd;
166:            char    comcod;
167:            char    errlow;
168:            char    errhigh;
169: };
170:
171: struct REQ_INI {
172:            char    reqlen;           /* リクエストヘッダの長さ       */
173:            char    unitcd;           /* ユニットコード               */
174:            char    comcod;           /* コマンドコード               */
175:            char    errlow;           /* エラーコードその1            */
176:            char    errhigh;          /* エラーコードその2            */
177:            char    rsv[8];           /* 予約領域                       */
178:                                      /* ---------------------------      */
179:            char    mxunit;           /* ユニット数                     */
180:            char    devend[4];        /* (*devend;)
181:                  デバイスドライバ制御プログラム終了アドレス */
182:            char    bpbpoi[4];        /* (struct       BPBPOI *bpbpoi;
183:                                                 BPBテーブルアドレス */
184:            char    bdevno;           /* ブロックデバイス番号(0=A*)    */
185: };
186:
187: struct REQ_CHG {
188:            char    reqlen;           /* リクエストヘッダの長さ       */
189:            char    unitcd;           /* ユニットコード               */
190:            char    comcod;           /* コマンドコード               */
191:            char    errlow;           /* エラーコードその1            */
192:            char    errhigh;          /* エラーコードその2            */
193:            char    rsv[8];           /* 予約領域                       */
194:                                      /* ---------------------------      */
195:            char    diskid;           /* よくわかんない<現在未使用>   */
196:            long    diskfg;           /* よくわかんない資料ではバイト型*/
197: };
198:
199: struct REQ_RW {
200:            char    reqlen;           /* リクエストヘッダの長さ       */
201:            char    unitcd;           /* ユニットコード               */
202:            char    comcod;           /* コマンドコード               */
203:            char    errlow;           /* エラーコードその1            */
204:            char    errhigh;          /* エラーコードその2            */
205:            char    rsv[8];           /* 予約領域                       */
206:                                      /* ---------------------------      */
207:            char    diskid;           /* よくわかんない<現在未使用>   */
208:            char    dmaadr[4];        /* (*dmaadr;) 転送バッファ       */
209:            long    dmalen;           /* 転送セクター数                */
210:            char    starec00[2];      /* アクセスセクタ番号上位2byte  */
211:            long    starec;           /* アクセスセクタ番号            */
212: };
213:
214: struct REQ_CTRL {
215:            char    reqlen;           /* リクエストヘッダの長さ       */
216:            char    unitcd;           /* ユニットコード               */
217:            char    comcod;           /* コマンドコード               */
218:            char    errlow;           /* エラーコードその1            */
219:            char    errhigh;          /* エラーコードその2            */
220:            char    rsv[8];           /* 予約領域                       */
221:                                      /* ---------------------------      */
222:            char    getdat;
223: };
224:
225: struct REQ_CHG req_hed;
226:
227: char       _rq_buf[1024];
228: char       _rw_buf[16384];
229: struct REQ_HED *_rq = (struct REQ_HED*)_rq_buf;
230: int        _drv;
231: int        _byte[10];
232: /*********************************************************************
233:            main() テスト用のメインルーチン
234:            エントリー(entry)は全プログラムの入り口と解釈すること
235: *********************************************************************/
236: void main( argc, argv )
237: int        argc;
238: unsigned char *argv[];
239: {
240:    int   sts;
241:    int   c;
242:    int   i;
243:     _rs_inz( BPS9600 );
244:     _drv = 4;              /* ドライブNoセット = D: */
245:     printf( "仮想DISK system V1.3 ¥n" );
246:     printf( "[ESC] exit ¥n" );
247:     for( i=1 ; i<argc ; i++ ) {
248:        if( memcmp( argv[i], "-d", 2 )==0 || memcmp( argv[i], "-D", 2
)==0 ) {
249:           _drv = (int)(argv[i][2] - 'A');
250:           if( argv[i][2] >= 'a' ) {
251:              _drv = (int)(argv[i][2] - 'a');
252:           }
253:           _drv ++;
254:        }
255:        else {
256:           printf( "R [-H] ¥n" );
257:           printf( "-H    ...... help message ¥n" );
258:           printf( "-D    ...... 仮想化開始のドライブを指定 ¥n" );
259:           printf( "             例: R -DC ¥n" );
260:           printf( "                Cドライブから仮想化する場合 ¥n" );
261:           exit(1);
262:        }
263:     }
264:     _rs_buf_clr();         /* 通信バッファクリア */
265:     xenable();
266:     while( 1 ) {
267:        if( _LOF232C() ) {        /* 同調機構、受信側でとりこぼし */
268:           xdisable();
269:           c = _INP232C();
270:           if( c!='S' ) {
271:              _rs_buf_clr();   /* 通信バッファクリア */
272:              _rs_buf_clr();   /* 通信バッファクリア */
273:           }
274:           else if( (sts=blk_out( "A", 1 )) ) {
275:              _rs_buf_clr();   /* 通信バッファクリア */
276:              _rs_buf_clr();   /* 通信バッファクリア */
277:           }
278:           else {
279:              rcv_ent();
280:           }
281:           xenable();
282:        }
283:        while( kbhit() ) {
284:           if( getch()==ESC ) {
285:              _rcv();
286:              exit(0);
287:           }
288:        }
289:     }
290: }
291:
292: /*
293: char       _rq_buf[1024];
294: char       _rw_buf[16384];
295: struct REQ_HED *_rq = (struct REQ_HED*)_rq_buf;
296: */
297: /****************************************
298:    rcv_ent 通信パケット処理
299: ****************************************/
300: void rcv_ent()
301: {
302:    int   sts;
303:    INT   n;
304:     if( (sts=blk_in_x( &n, sizeof(n)  ))) {    /* パケット長取得 */
305:         ;
306:     }
307:     else if( (sts=blk_in( _rq, n  ))) {    /* パケット取得    */
308:         ;
309:     }
310:     switch( _rq->comcod ) {
311:        case 0:
312:           sts = r_dskini( _rq );
313:           break;
314:        case 1:
315:           sts = r_mediac( _rq );
316:           break;
317:        case 2:
318:           sts = r_notcom( _rq );    /* 未使用命令 */
319:           break;
320:        case 3:
321:           sts = r_notcom( _rq );    /* 未使用命令 */
322:           break;
323:        case 4:
324:           sts = r_dskinp( _rq );
325:           break;
326:        case 5:
327:           sts = r_dskctrl( _rq );
328:           break;
329:        case 6:
330:           sts = r_notcom( _rq );    /* 未使用命令 */
331:           break;
332:        case 7:
333:           sts = r_notcom( _rq );    /* 未使用命令 */
334:           break;
335:        case 8:
336:           sts = r_dskout( _rq );
337:           break;
338:        case 9:
339:           sts = r_dskotv( _rq );
340:           break;
341:        case 10:
342:           sts = r_notcom( _rq );    /* 未使用命令 */
343:           break;
344:        case 11:
345:           sts = r_notcom( _rq );    /* 未使用命令 */
346:           break;
347:        case 12:
348:           sts = r_notcom( _rq );    /* 未使用命令 */
349:           break;
350:        default:
351:           sts = r_notcom( _rq );    /* 未使用命令 */
352:           break;
353:     }
354: }
355:
356: /********************************
357:    notcom  未使用命令
358: ********************************/
359: int r_notcom(req)
360: struct REQ_HED *req;
361: {
362:     req->errlow = 0x03;     /* 下位バイトエラーコード格納   */
363:     req->errhigh = 0x50;    /* 上位バイトエラーコード格納   */
364:     return( 0 );
365: }
366:
367: /*****************************************************
368:   r_mediac メディア交換処理(RAMなので未使用のはず)
369: *****************************************************/
370: int r_mediac( req_chg )
371: struct REQ_CHG *req_chg;
372: {
373:    int   sts;
374:    int   n;
375:     req_chg->diskfg = 0L;    /* stsコードセット */
376:          /* リクエストヘッダ送信 */
377:     if( (sts=blk_out( &(req_chg->diskfg), sizeof(req_chg->diskfg) ))
```

▶私のX68000は昼間はなんともないが，夜になるとファンが唸りを上げて動く。天変地異の前触れだろうか。

宝福　公司(26)北海道

```
) {
 378:          ;
 379:      }
 380:      return( 0 );
 381: }
 382:
 383: /********************************
 384:   r_dskinp ディスク入力処理(?)
 385: ********************************/
 386: int r_dskinp( req )
 387: struct REQ_RW *req;
 388: {
 389:    int   sts;
 390:    int   n;
 391:    long   len;
 392:    long   rec;
 393:    long   wk;
 394:    long   wkl;
 395:    char   *p;
 396:     req->errlow = 0;  /* 下位バイトエラーコード格納 */
 397:     req->errhigh = 0; /* 上位バイトエラーコード格納 */
 398:     XCHG4( &(req->dmalen) ); /* もしDOSなら配列変換 */
 399:     XCHG4( &(req->starec) ); /* もしDOSなら配列変換 */
 400:     len = req->dmalen * (long)_byte[req->unitcd];
 401:     rec =req->starec;
 402:     rec >>= 16;
 403:     if( (sts=blk_out_x( &len, 4 ))) { /* 転送データ長 */
 404:        ;
 405:     }
 406:     else {
 407:        len = req->dmalen * (long)_byte[req->unitcd];
 408:        while( len > 0 ) {
 409:           wk = len;
 410:           if( wk > 16384L ) {
 411:              wk = 16384L;
 412:           }
 413:           wkl = wk / (long)_byte[req->unitcd];
 414:           DISKREDX( (unsigned char*)_rw_buf, req->unitcd+_drv, rec,
wkl );
 415:           if( (sts=blk_out( _rw_buf, wk ))) { /* データ長送信 */
 416:              break;
 417:           }
 418:           rec += wkl;
 419:           len -= 16384L;
 420:        }
 421:     }
 422:     return( sts );
 423: }
 424:
 425: /****************************************
 426:   r_dskotv 出力&ベリファイ処理(?)
 427:   ASMを強引にハンドディスコンパイル
 428: ****************************************/
 429: int r_dskotv( req )
 430: struct REQ_RW *req;
 431: {
 432:     return( r_dskout( req ) );
 433: }
 434:
 435: /********************************
 436:   r_dskout ディスク出力処理(?)
 437: ********************************/
 438: int r_dskout( req )
 439: struct REQ_RW *req;
 440: {
 441:    int   sts;
 442:    long   len;
 443:    long   rec;
 444:    long   wk;
 445:     req->errlow = 0;     /* 下位バイトエラーコード格納 */
 446:     req->errhigh = 0;    /* 上位バイトエラーコード格納 */
 447:     XCHG4( &(req->dmalen) ); /* もしDOSなら配列変換 */
 448:     XCHG4( &(req->starec) ); /* もしDOSなら配列変換 */
 449:     rec =req->starec;
 450:     rec >>= 16;
 451:     if( (sts=blk_in_x( &len, 4 ))) { /* データ長受信 */
 452:        ;
 453:     }
 454:     else {
 455:        while( len > 0 ) {
 456:           wk = len;
 457:           if( wk > 16384L ) {
 458:              wk = 16384L;
 459:           }
 460:           if( (sts=blk_in( _rw_buf, wk ))) { /* データ長送信 */
 461:              break;
 462:           }
 463:           wk /= (long)_byte[req->unitcd];
 464:           DISKWRTX( (unsigned char*)_rw_buf, req->unitcd+_drv, rec,
wk );
 465:           _rs_buf_clr();
 466:           _OUT232C('A');
 467:           rec += 16384L / (long)_byte[req->unitcd];
 468:           len -= 16384L;
 469:        }
 470:        if( len<=0 ) {
 471:           sts=blk_out( &(req->errlow), 2 );
 472:        }
 473:     }
 474:
 475:     return( sts );
 476: }
 477:
 478: /**********************************
 479:   r_dskctrl ディスクコントロール
 480: **********************************/
 481: int r_dskctrl( req )
 482: struct REQ_CTRL *req;
 483: {
 484:    int   sts;
 485:    int   n;
 486:    long   mode;
 487:     req->errlow = 0;  /* 下位バイトエラーコード格納 */
 488:     req->errhigh = 0; /* 上位バイトエラーコード格納 */
 489:     n = (int)(req->reqlen);
 490:     mode = 0L;
 491:     mode = DRVCTRL( mode, (int)(req->unitcd)+_drv );
 492:     req->getdat = (char)mode;
 493:     if( (sts=blk_out( req, n ))) { /* リクエストヘッダ送信 */
 494:        ;
 495:     }
 496:     else {
 497:        sts = 0;
 498:     }
 499:     return( sts );
 500: }
 501: struct   BPB_TBL   {
 502:    unsigned short b_no;          /* セクタあたりのバイト数 */
 503:    unsigned char  sct_no;        /* クラスタあたりのセクタ数 */
 504:    unsigned char  fat_no;        /* ファット領域の個数 */
 505:    unsigned short rsv_sct_no;    /* 予約領域のセクタ数 */
 506:    unsigned short root_ent_no;   /* ルートの最大ファイル数 */
 507:    unsigned short sct_max;       /* 全セクター数  */
 508:    unsigned char  id;            /* メディアバイト */
 509:    unsigned char  fat_sct_no;    /* 1fatのセクタ数 */
 510: };
 511:
 512: /********************************
 513:   dskini ディスク初期化ルーチン
 514: ********************************/
 515: int r_dskini( req )
 516: struct REQ_INI *req;
 517: {
 518:    int   sts;
 519:    struct   DPBPTR   d;
 520: #if SYSTEM==PC98
 521:    struct         DPBPTR1 *d1;
 522: #endif
 523:    struct   BPB_TBL   bpb_tbl;
 524:    int   fat;
 525:    int   fat_no;
 526:    int   rsv_sct;
 527:    char  dsk_flg;
 528:    int   drv;
 529:    char  d_no;
 530:    long  mode;
 531:    long  old_drv;
 532:    long  wk_drv;
 533:    int   ver;
 534:    int   sub_ver;
 535:
 536: #if SYSTEM==X68
 537:       _OUT232C( BPS76800 );
 538:      dlytime();
 539:      _rs_spd( BPS76800 );
 540:      printf( "Speed up to 76,800bps¥n¥r" );
 541:      dlytime();
 542: #endif
 543: #if SYSTEM==PC98
 544:       _OUT232C( BPS9600 );
 545:      dlytime();
 546: /*     _rs_spd( BPS9600 );
 547:      dlytime();*/
 548: #endif
 549:      drv = _drv;
 550:      old_drv = CURDRV(); /* 現在のドライブ保存 */
 551:      while( 1 ) {
 552:         CHGDRV( (long)drv-1 );
 553:         wk_drv = CURDRV();
 554:         wk_drv++;
 555:         if( drv != wk_drv ) {
 556:            dsk_flg = -1;
 557:            sts = blk_out( &dsk_flg, sizeof(dsk_flg) );
 558:            break;
 559:         }
 560:         mode = 0L;
 561:         mode = DRVCTRL( mode, drv );
 562:         mode &= 2L;
 563:         d_no = 'A' + (char)drv;
 564:         d_no --;
 565:         sts = GETDPB( drv, &d );
 566:         if( mode==0 ) {
 567:            bpb_tbl.b_no = 1024;          /* セクタあたりのバイト数 */
 568:            bpb_tbl.sct_no = 1;           /* クラスタあたりのセクタ数 */
 569:            bpb_tbl.fat_no = 2;           /* ファット領域の個数 */
 570:            bpb_tbl.rsv_sct_no = 1;       /* 予約領域のセクタ数 */
 571:            bpb_tbl.root_ent_no = 192;    /* ルートの最大ファイル数 */
 572:            bpb_tbl.sct_max = 1232;       /* 全セクター数 */
 573:            bpb_tbl.id = 0xfe;            /* メディアバイト */
 574:            bpb_tbl.fat_sct_no = 2;       /* 1fatのセクタ数 */
 575:            printf( "%c: を主機へ仮想ドライブとして登録. DISKが未セットなので2HDに設定.
¥n", d_no  );
 576:         }
 577:         else if( d.id==0xfe ) {
 578:            bpb_tbl.b_no = 1024;          /* セクタあたりのバイト数 */
 579:            bpb_tbl.sct_no = 1;           /* クラスタあたりのセクタ数 */
 580:            bpb_tbl.fat_no = 2;           /* ファット領域の個数 */
 581:            bpb_tbl.rsv_sct_no = 1;       /* 予約領域のセクタ数 */
 582:            bpb_tbl.root_ent_no = 192;    /* ルートの最大ファイル数 */
 583:            bpb_tbl.sct_max = 1232;       /* 全セクター数 */
 584:            bpb_tbl.id = 0xfe;            /* メディアバイト */
 585:            bpb_tbl.fat_sct_no = 2;       /* 1fatのセクタ数 */
 586:            printf( "%c: を主機へ仮想ドライブとして登録 ¥n", d_no );
 587:         }
 588:         else {
 589: #if SYSTEM==X68
 590:            bpb_tbl.b_no = d.byte;            /* セクタあたりのバイト数 */
 591:            bpb_tbl.sct_no = d.sec + 1;       /* クラスタあたりのセクタ数 */
 592:            bpb_tbl.fat_no = d.fatcount;      /* ファット領域の個数 */
 593:            bpb_tbl.rsv_sct_no = d.fatsec; /* 予約領域のセクタ数 */
 594:            bpb_tbl.root_ent_no = d.dircount;  /* ルートの最大ファイル数 */
 595:            bpb_tbl.sct_max = d.maxfat*(d.sec+1); /* 全セクター数 */
 596:            bpb_tbl.id = d.id;                /* メディアバイト */
 597:            bpb_tbl.fat_sct_no = d.fatlen; /* 1fatのセクタ数 */
 598:            printf( "%c: を主機へ仮想ドライブとして登録 ¥n", d_no );
 599: #endif
 600: #if SYSTEM==PC98
 601:        getver( &ver, &sub_ver );
 602:        if( ver<4 ) {
 603:         bpb_tbl.b_no = d.byte;            /* セクタあたりのバイト数 */
 604:         bpb_tbl.sct_no = d.sec + 1;       /* クラスタあたりのセクタ数 */
 605:         bpb_tbl.fat_no = d.fatcount;      /* ファット領域の個数 */
 606:         bpb_tbl.rsv_sct_no = d.fatsec; /* 予約領域のセクタ数 */
 607:         bpb_tbl.root_ent_no = d.dircount; /* ルートの最大ファイル数 */
 608:         bpb_tbl.sct_max = d.maxfat*(d.sec+1); /* 全セクター数 */
 609:         bpb_tbl.id = d.id;                /* メディアバイト */
 610:         bpb_tbl.fat_sct_no = d.fatlen; /* 1fatのセクタ数 */
 611:         printf( "%c: を主機へ仮想ドライブとして登録 ¥n", d_no );
 612:       }
 613:         else {
 614:         d1 = (struct DPBPTR1*)(&d);
 615:         bpb_tbl.b_no = d1->byte;           /* セクタあたりのバイト数 */
 616:         bpb_tbl.sct_no = d1->sec + 1;      /* クラスタあたりのセクタ数 */
 617:         bpb_tbl.fat_no = d1->fatcount;     /* ファット領域の個数 */
 618:         bpb_tbl.rsv_sct_no = d1->fatsec; /* 予約領域のセクタ数 */
 619:         bpb_tbl.root_ent_no = d1->dircount; /* ルートの最大ファイル数 */
 620:         bpb_tbl.sct_max = d1->maxfat*(d1->sec+1); /* 全セクター数 */
 621:         bpb_tbl.id = d1->id;               /* メディアバイト */
 622:         bpb_tbl.fat_sct_no = d1->fatlen; /* 1fatのセクタ数 */
 623:         printf( "%c: を主機へ仮想ドライブとして登録 ¥n", d_no );
 624:       }
 625: #endif
 626:         }
 627:         dsk_flg = 0;
 628:         if( (sts=blk_out( &dsk_flg, sizeof(dsk_flg) ))) { /* bpb tbl
送信 */
 629:            break;
 630:         }
 631:         else {
 632:            XCHG2( &(bpb_tbl.b_no) );          /* もしDOSなら配列変換 */
 633:            XCHG2( &(bpb_tbl.rsv_sct_no) );   /* もしDOSなら配列変換 */
 634:            XCHG2( &(bpb_tbl.root_ent_no) ); /* もしDOSなら配列変換 */
 635:            XCHG2( &(bpb_tbl.sct_max) );      /* もしDOSなら配列変換 */
 636:            if( (sts=blk_out( &bpb_tbl, sizeof( bpb_tbl ) ))) { /* bpb
tbl 送信 */
 637:               break;
 638:            }
 639:            _byte[drv-_drv] = bpb_tbl.b_no;
 640:            drv ++;
 641:         }
 642:      }
 643:      CHGDRV( old_drv );        /* ドライブ復旧 */
 644:      return( sts );
 645: }
 646:
 647: /********************************
 648:     PC9801機種依存関数
 649: ********************************/
 650: #if SYSTEM==PC98
 651:
 652: char        __dmy[128];
 653: /**************************
 654:   _rs_inz       初期化
 655:  return   :   sts
 656: **************************/
 657: int        _rs_inz( spd )
 658: int    spd;
 659: {
 660:    union   REGS    i_reg;
 661:    union   REGS    o_reg;
 662:    struct  SREGS   s_reg;
 663:    int    sts;
 664:     segread( &s_reg );
 665:     i_reg.h.ah = 0;
 666:     i_reg.h.al = 0x07;
 667:     i_reg.h.ch = 0x4e;
 668:     i_reg.h.cl = 0x37;
 669:     s_reg.es = s_reg.ds;
 670:     i_reg.x.di = (int)__dmy;
 671:     i_reg.x.dx = 128;
 672:     i_reg.h.bh = 0x2;
 673:     i_reg.h.bl = 0x1e;
 674:     int86x( 0x19, &i_reg, &o_reg, &s_reg );
 675:     return( i_reg.h.ah );
 676: }
 677:
 678: /****************************************
 679:   _LOF232C 受信バッファチェックルーチン
 680: ****************************************/
 681: int    _LOF232C()
 682: {
 683:    union   REGS   i_reg;
 684:    union   REGS    o_reg;
 685:    int   sts;
 686:     i_reg.h.ah = 2;
 687:     i_reg.h.al = 0;
 688:     int86( 0x19, &i_reg, &o_reg );
 689:     sts = o_reg.x.cx;
 690:     return( sts );
 691: }
 692:
 693: /********************************
 694:   _INP232C 1文字受信ルーチン
 695: ********************************/
```

▶パソコン通信を始めて，X68000の前にいる時間がまた長くなった。

田高　浩一(28)東京都

```
696: int    _INP232C()
697: {
698:   union   REGS   i_reg;
699:   union   REGS   o_reg;
700:   unsigned int   sts;
701:    i_reg.h.ah = 4;
702:    i_reg.h.al = 0;
703:    int86( 0x19, &i_reg, &o_reg );
704:    sts = o_reg.h.ch;
705:    return( (int)sts );
706: }
707:
708: /******************************
709:  _OUT232C 1文字送信ルーチン
710: ******************************/
711: void   _OUT232C( int c )
712: {
713:   union   REGS   i_reg;
714:   union   REGS   o_reg;
715:   unsigned int   sts;
716:    i_reg.h.ah = 3;
717:    i_reg.h.al = (unsigned char)(c & 0xff);
718:    int86( 0x19, &i_reg, &o_reg );
719: }
720:
721: /************************************
722:  GETDPB DPBテーブル取得ルーチン
723: ************************************/
724: int      GETDPB( int drv, struct DPBPTR  *d )
725: {
726:   union   REGS   i_reg;
727:   union   REGS   o_reg;
728:   struct  SREGS  s_reg;
729:   unsigned int   sts;
730:   unsigned int   ds;
731:    segread( &s_reg );
732:    ds = s_reg.ds;
733:    i_reg.h.ah = 0x32;
734:    i_reg.h.dl = drv;
735:    int86x( 0x21, &i_reg, &o_reg, &s_reg );
736:    movedata( s_reg.ds, o_reg.x.bx, ds, d, sizeof(struct DPBPTR) );
737:    sts = o_reg.h.al;
738:    return( sts );
739: }
740: /*---------------------- henkou 94/8/14 --------------*/
741: /*********************************
742:  DISKREDX ディスクリードルーチン
743: *********************************/
744: void DISKREDX( unsigned char* buf, int drv, long rec, long len )
745: {
746:   unsigned int   sts;
747:   int    i;
748:    for( i=0 ; i<5 ; i++ ) {
749:       sts = diskred( buf, drv, rec, len );
750:       if( sts==0 ) {
751:          break;
752:       }
753:       else if( sts==2 ) {
754:          printf( "¥aディスクをセットして、どれかキーを押してください。" );
755:       }
756:       else {
757:          printf( "¥aディスクに障害があります。確認後、どれかキーを押してください。" );
758:       }
759:       getch();
760:       printf( "¥n" );
761:    }
762: }
763:
764: /*********************************
765:  diskred ディスクリードルーチン
766: *********************************/
767: int diskred( unsigned char* buf, int drv, long rec, long len )
768: {
769:   union   REGS   i_reg;
770:   union   REGS   o_reg;
771:   struct  SREGS  s_reg;
772:   unsigned int   sts;
773:   unsigned int   ds;
774:    drv --;
775:    segread( &s_reg );
776:    ds = s_reg.ds;
777:    i_reg.h.al = drv;
778:    i_reg.x.bx = (unsigned int)buf;
779:    i_reg.x.cx = len;
780:    i_reg.x.dx = rec;
781:    int86x( 0x25, &i_reg, &o_reg, &s_reg );
782:    dlytime();
783:    sts = o_reg.x.ax;
784:    sts &= 0x000f;
785:    return( sts );
786: }
787:
788: /***********************************
789:  DISKWRTX ディスクライトルーチン
790: ***********************************/
791: void DISKWRTX( unsigned char* buf, int drv, long rec, long len )
792: {
793:   union   REGS   i_reg;
794:   union   REGS   o_reg;
795:   struct  SREGS  s_reg;
796:   unsigned int   sts;
797:   unsigned int   ds;
798:   int    i;
799:    for( i=0 ; i<5 ; i++ ) {
800:       sts = diskwrt( buf, drv, rec, len );
801:       if( sts==0 ) {
802:          break;
803:       }
804:       else if( sts==2 ) {
805:          printf( "¥aディスクをセットして、どれかキーを押してください。" );
806:       }
807:       else {
808:          printf( "¥aディスクに障害があります。確認後、どれかキーを押してください。" );
809:       }
810:       getch();
811:       printf( "¥n" );
812:    }
813: }
814:
815: /*********************************
816:  diskwrt ディスクライトルーチン
817: *********************************/
818: int diskwrt( unsigned char* buf, int drv, long rec, long len )
819: {
820:   union   REGS   i_reg;
821:   union   REGS   o_reg;
822:   struct  SREGS  s_reg;
823:   unsigned int   sts;
824:   unsigned int   ds;
825:    drv --;
826:    segread( &s_reg );
827:    ds = s_reg.ds;
828:    i_reg.h.al = drv;
829:    i_reg.x.bx = (unsigned int)buf;
830:    i_reg.x.cx = len;
831:    i_reg.x.dx = rec;
832:    int86x( 0x26, &i_reg, &o_reg, &s_reg );
833:    dlytime();
834:    sts = o_reg.x.ax;
835:    sts &= 0x000f;
836:    return( sts );
837: }
838:
839: /*************************************
840:  DRVCTRL    ドライブ情報取得ルーチン
841: *************************************/
842: long DRVCTRL( long mode, int drv )
843: {
844:   union   REGS   i_reg;
845:   union   REGS   o_reg;
846:   unsigned int   sts;
847:   char wk[16384];
848:    i_reg.h.ah = 0x1c;
849:    i_reg.h.al = drv-1;
850:    i_reg.x.bx = (unsigned int)(wk);
851:    i_reg.x.cx = 1;
852:    i_reg.x.dx = 1;
853:    int86( 0x25, &i_reg, &o_reg );
854:    sts = 0x000f & o_reg.h.al;
855:    if( sts==2 ) {
856:       sts = 0;
857:    }
858:    else {
859:       sts = 2;
860:    }
861:    return( sts );
862: }
863:
864: /*************************************
865:  CURDRV 現在のドライブ取得ルーチン
866: *************************************/
867: long CURDRV()
868: {
869:   unsigned int   drv;
870:    _dos_getdrive( &drv );
871:    drv--;
872:    return( (long)drv );
873: }
874:
875: /************************************
876:  CHGDRV 現在のドライブ変更ルーチン
877: ************************************/
878: long CHGDRV( long drv )
879: {
880:   unsigned int   drv_max;
881:    drv++;
882:    _dos_setdrive( (int)drv, &drv_max );
883:    return( (int)drv_max );
884: }
885: /*---------------------- henkou 94/8/14 end -------------/
886: /**********************************************
887:  blk_in_x ブロック受信、受信データを並べ替え
888: **********************************************/
889: int blk_in_x( unsigned char *data, int len )
890: {
891:   int    sts;
892:    sts = blk_in( data, len );
893:    xchg4( data );
894: }
895:
896: /**********************************************
897:  blk_out_x ブロック送信、送信データを並べ替え
898: **********************************************/
899: int blk_out_x( unsigned char *data, int len )
900: {
901:   int    sts;
902:   char    wk[4];
903:    wk[3] = data[0];
904:    wk[2] = data[1];
905:    wk[1] = data[2];
906:    wk[0] = data[3];
907:    sts = blk_out( wk, len );
908: }
909:
910: /*********************************
911:  xchg2  2バイトデータの並べ替え
912: *********************************/
913: void xchg2( char *data )
914: {
915:   char    wk[2];
916:    wk[0] = data[1];
917:    wk[1] = data[0];
918:    data[0] = wk[0];
919:    data[1] = wk[1];
920: }
921:
922: /*********************************
923:  xchg4  4バイトデータの並べ替え
924: *********************************/
925: void xchg4( char *data )
926: {
927:   char    wk[4];
928:    wk[0] = data[3];
929:    wk[1] = data[2];
930:    wk[2] = data[1];
931:    wk[3] = data[0];
932:    data[0] = wk[0];
933:    data[1] = wk[1];
934:    data[2] = wk[2];
935:    data[3] = wk[3];
936: }
937: void xenable()
938: {
939:    _enable();
940: }
941: void xdisable()
942: {
943:    _disable();
944: }
945:
946: /*********************************
947:  getver DOS Version 取得ルーチン
948: *********************************/
949: void getver( int *ver, int *sub_ver )
950: {
951:   union   REGS   i_reg;
952:   union   REGS   o_reg;
953:   struct  SREGS  s_reg;
954:    i_reg.h.ah = 0x30;
955:    i_reg.h.al = 0;
956:    int86x( 0x21, &i_reg, &o_reg, &s_reg );
957:    *ver = o_reg.h.al;
958:    *sub_ver = o_reg.h.ah;
959: }
960:
961: /**********************************
962:  _rcv()     通信割り込み設定の復旧
963:  return    : char, or sts
964: **********************************/
965: void _rcv()
966: {
967: }
968: #endif
969: #if SYSTEM==X68
970:
971: /***********************************
972:  DISKREDX    ディスクリードルーチン
973: ***********************************/
974: void DISKREDX( unsigned char* buf, int drv, long rec, long len )
975: {
976:   unsigned int   sts;
977:   int    i;
978:   long    mode;
979:    while( 1 ) {
980:       mode = 0L;
981:       mode = DRVCTRL( mode, drv );
982:       mode &= 2L;
983:       if( mode ) {
984:          DISKRED( buf, drv, rec, len );
985:          break;
986:       }
987:       printf( "¥aディスクをセットして、どれかキーを押してください。" );
988:       getch();
989:       printf( "¥n" );
990:    }
991: }
992:
993: /***********************************
994:  DISKWRTX    ディスクライトルーチン
995: ***********************************/
996: void DISKWRTX( unsigned char* buf, int drv, long rec, long len )
997: {
998:   unsigned int   sts;
999:   int    i;
1000:   long    mode;
1001:    while( 1 ) {
1002:       mode = 0L;
1003:       mode = DRVCTRL( mode, drv );
1004:       mode &= 2L;
1005:       if( mode ) {
1006:          DISKWRT( buf, drv, rec, len );
1007:          break;
1008:       }
1009:       printf( "¥aディスクをセットして、どれかキーを押してください。" );
1010:       getch();
1011:       printf( "¥n" );
1012:    }
1013: }
1014: /* ======== 1994.12.13 修正 =========*/
1015: int    __mode;
1016: int    _spd_tbl[] = { 128, 63, 31, 14, 6, 2, 0 };
1017: BYTE    int_a_sts;
1018: BYTE    int_b_sts;
1019: /**********************
```

▶ 2月号の吉本さんへ。2週間で14ならまだいい。私の場合は1週間で13もの試験。もうなにから手をつけたらいいのやら……。　古川　研二(21)埼玉県

```
1020:   _rs_inz        初期化
1021:   return    :    sts
1022: ****************************/
1023:  _rs_inz( spd )
1024: int   spd;       /* speed    */
1025: {
1026:      __mode = SET232C( -1 );
1027:     SET232C( 0x4c07 );
1028:     B_BPOKE(0xe98005, 0x03);    /* 受信モードセット */
1029:     B_BPOKE(0xe98005, 0xe1);
1030:     B_BPOKE(0xe98005, 0x05);    /* 送信モードセット */
1031:     B_BPOKE(0xe98005, 0xea);
1032:     B_BPOKE(0xe98005, 0x0c);    /* ボーレートセット */
1033:     B_BPOKE(0xe98005,  _spd_tbl[spd] );
1034:     B_BPOKE(0xe98005, 0x0d);
1035:     B_BPOKE(0xe98005, 0x00);
1036:     B_BPOKE(0xe98005, 0x01);    /* ポーリングモードセット */
1037:     B_BPOKE(0xe98005, 0x01);
1038:     int_a_sts = B_BPEEK( 0xe88007 );
1039:     int_b_sts = B_BPEEK( 0xe88009 );
1040: /*    B_BPOKE( 0xe88007, 0x1e );
1041:     B_BPOKE( 0xe88009, 0x00 );
1042: */
1043:     return( 0 );
1044: }
1045:
1046: /***************************
1047:  _rs_spd        速度変更
1048:   return    :    sts
1049: ****************************/
1050:  _rs_spd( spd )
1051: int   spd;       /* speed */
1052: {
1053:     B_BPOKE(0xe98005, 0x03);    /* 受信モードセット */
1054:     B_BPOKE(0xe98005, 0xe1);
1055:     B_BPOKE(0xe98005, 0x05);    /* 送信モードセット */
1056:     B_BPOKE(0xe98005, 0xea);
1057:     B_BPOKE(0xe98005, 0x0c);    /* ボーレートセット */
1058:     B_BPOKE(0xe98005,  _spd_tbl[spd] );
1059:     B_BPOKE(0xe98005, 0x0d);
1060:     B_BPOKE(0xe98005, 0x00);
1061:     B_BPOKE(0xe98005, 0x01);    /* ポーリングモードセット */
1062:     B_BPOKE(0xe98005, 0x01);
1063:     return( 0 );
1064: }
1065:
1066: /************************************
1067:   _rcv()    通信割り込み設定の復旧
1068:   return    : char, or sts
1069: ************************************/
1070: void _rcv()
1071: {
1072:     B_BPOKE(0xe98005, 0x01);
1073:     B_BPOKE(0xe98005, 0x11);
1074:     B_BPOKE( 0xe88007, int_a_sts );
1075:     B_BPOKE( 0xe88009, int_b_sts );
1076:     SET232C( __mode );
1077: }
1078:
1079: /*******************************************
1080:   _LOF232C    受信バッファチェックルーチン
1081: *******************************************/
1082: int    _LOF232C()
1083: {
1084:   long   i;
1085:   int    sts;
1086:    sts = 0;
1087:    B_BPOKE(0xe98005, 0);
1088:     if( B_BPEEK(0xe98005)&0x01 ) {
1089:        sts = 1;
1090:     }
1091:     return( sts );
1092: }
1093:
1094: /*********************************
1095:   _INP232C    1文字受信ルーチン
1096: *********************************/
1097: int    _INP232C()
1098: {
1099:   long   i;
1100:   int    sts;
1101:    sts = -1;
1102:    for( i=0 ; i<1000000L ; i++ ) {
1103:       B_BPOKE(0xe98005, 0);
1104:       if( B_BPEEK(0xe98005)&0x01 ) {
1105:          sts = B_BPEEK(0xe98007);
1106:          break;
1107:       }
1108:    }
1109:    return( sts );
1110: }
1111:
1112: /*********************************
1113:   _OUT232C    1文字送信ルーチン
1114: *********************************/
1115: void    _OUT232C( int c )
1116: {
1117:   long   i;
1118:   int    sts;
1119:    sts = -1;
1120:    for( i=0 ; i<1000000L ; i++ ) {
1121:       B_BPOKE(0xe98005, 0);
1122:       if( B_BPEEK(0xe98005)&0x04 ) {
1123:          B_BPOKE(0xe98007, (unsigned char)c);    /* 1文字送出 */
1124:          sts = 0;
1125:          break;
1126:       }
1127:    }
1128: }
1129: void xenable()
1130: {
1131:    _rcv1();
1132: }
1133: void xdisable()
1134: {
1135:    B_BPOKE( 0xe88007, 0x1e );
1136:    B_BPOKE( 0xe88009, 0x00 );
1137: }
1138: void _rcv1()
1139: {
1140:    B_BPOKE( 0xe88007, int_a_sts );
1141:    B_BPOKE( 0xe88009, int_b_sts );
1142: }
1143: #endif
1144:
1145: void dlytime()
1146: {
1147:   long   i;
1148:   long   a;
1149:   long   b;
1150:    for( i=1; i<30000 ; i++ ) {
1151:       a=100;
1152:       b=200;
1153:       a=a*b;
1154:    }
1155: }
```

リスト5 D3.C

```
  1: #define          X68     0
  2: #define          PC98    1
  3: #define          IBM     2
  4: #define          SYSTEM  0
  5:
  6: #if SYSTEM==X68
  7: #include         <doslib.h>
  8: #include         <stdio.h>
  9: #include         <time.h>
 10: #endif
 11:
 12: #if SYSTEM==PC98 || SYSTEM==IBM
 13: #include         <stdlib.h>
 14: #include         <dos.h>
 15: #include         <sys¥types.h>
 16: #include         <sys¥stat.h>
 17: #define          NULL    0
 18: #endif
 19:
 20: #define TIMEOUT 5L
 21:
 22: int      blk_in();
 23: int      blk_in1();
 24: int      blk_out();
 25: int      blk_out1();
 26: void     _rs_buf_clr();
 27:
 28: int      _LOF232C();
 29: int      _INP232C();
 30: void     _OUT232C();
 31:
 32: *****************************************
 33:  blk_in    複数データを受信
 34:     受信に失敗したら5回までやりなおす
 35: *****************************************/
 36: int   blk_in( data, len )
 37: unsigned char   *data;    /* 転送データ格納アドレス */
 38: int   len;                /* 転送データ長 */
 39: {
 40:   int   sts;
 41:   int   i;
 42:    for( i=0 ; i<5 ; i++ ) {
 43:       sts = blk_in1( data, len );
 44:       if( sts==0 ) {
 45:          break;
 46:       }
 47:    }
 48:    return( sts );
 49: }
 50:
 51: /******************************
 52:    blk_in1    複数データ受信
 53: ******************************/
 54: int   blk_in1( data, len )
 55: unsigned char   *data; /* 転送データ格納アドレス */
 56: int   len;             /* 転送データ長 */
 57: {
 58:   time_t   tm;
 59:   time_t   tm1;
 60:   time_t   tmx;
 61:   int   bsc;
 62:   int   i;
 63:   unsigned char   *ptr;
 64:   int   c;
 65:   int   sts;
 66:   int   n;
 67:    sts = -1;
 68:    bsc = 0;
 69:    ptr = (unsigned char *)data;
 70:    for( i=0 ; i<len ; i++ ) {
 71:       c = _INP232C();
 72:       *ptr++ = c;
 73:       bsc ^= c;
 74:    }
 75:    c = _INP232C();
 76:    _rs_buf_clr();
 77:    if( c!=bsc ) {
 78:       _OUT232C( 1 );
 79:    }
 80:    else {
 81:       _OUT232C( 0 );
 82:       sts = 0;
 83:    }
 84:    return( sts );
 85: }
 86:
 87: /****************************************
 88:    blk_out    複数データを送信
 89:    送信に失敗したら5回までやりなおす
 90: ****************************************/
 91: int   blk_out( data, len )
 92: unsigned char   *data; /* 転送データ格納アドレス */
 93: int   len;             /* 転送データ長 */
 94: {
 95:   int   sts;
 96:   int   i;
 97:    for( i=0 ; i<5 ; i++ ) {
 98:       _rs_buf_clr();     /* 通信バッファクリア */
 99:       sts = blk_out1( data, len );
100:       if( sts==0 ) {
101:          break;
102:       }
103:    }
104:    return( sts );
105: }
106:
107: /******************************
108:    blk_out1    複数データ送信
109: ******************************/
110: int   blk_out1( data, len )
111: unsigned char   *data; /* 転送データ格納アドレス */
112: int   len;             /* 転送データ長 */
113: {
114:   time_t   tm;
115:   time_t   tm1;
116:   time_t   tmx;
117:   int   bsc;
118:   int   i;
119:   unsigned char   *ptr;
120:   int   c;
121:   int   sts;
122:   int   n;
123:    sts = -1;
124:    bsc = 0;
125:    ptr = (unsigned char*)data;
126:    for( i=0 ; i<len ; i++ ) {
127:       if( _LOF232C() ) { /* 同期機構、受信側でとりこぼし */
128:          return(sts);     /* 強制リターン */
129:       }
130:       c = *ptr;
131:       _OUT232C( c );
132:       bsc ^= c;
133:       ptr ++;
134:    }
135:    _OUT232C( bsc );
136:    tm = time( NULL );
137:    while(1) {
138:       tm1 = time( NULL );        /* TimeOut チェック */
139:       tmx = tm1 - tm;
140:       if( tmx> TIMEOUT ) {
141:          break;
142:       }
143:       if( _LOF232C() ) {
144:          c = _INP232C();
145:          if( c==0 ) {
146:             sts = 0;
147:          }
148:          break;
149:       }
150:    }
151:    return( sts );
152: }
153:
154: /****************************************
155:    _rs_buf_clr    通信バッファクリア
156: ****************************************/
157: void _rs_buf_clr()
158: {
159:   int   c;
160:   int   i;
161:    for( i=0 ; i<200 ; i++ ) {
162:       while( _LOF232C() ) {
163:          c = _INP232C();
164:       }
165:    }
166: }
```

〈対応機種一覧〉●MZ-80K/C/700/1500●MZ-80B/2000●MZ-2500/2861●X1●X1turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA/7●FM-7/77/AV●MSX/2/2+/turbo R●PC-286/386/486/9801/98/9821●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。

第153部　S-OSシステムコールライブラリ(SOSLIB)

●S-OSシステムコールライブラリ

久しぶりの本編は，木下氏制作のS-OSシステムコールライブラリ(SOSLIB)を掲載します。これはその名のとおり，Small-CからS-OS"SWORD"のシステムコールの呼び出しを行う関数群です。

いままで，Small-CからS-OS"SWORD"システムを呼び出すには，アセンブラコードを使うことで対処するしかありませんでした。といっても，S-OS"SWORD"のシステムサブルーチンを呼び出すには必要な引数をレジスタに渡して，サブルーチンコールするだけというお手軽さ。いわば，ライブラリなどなくてもなんとかなる状態だったのです。しかし，C言語のソースにアセンブラニーモニックを書き込む煩わしさは，ないほうがいいに決まっています。Small-CからS-OSを手軽に扱いたい，という方はこのライブラリを利用してください。また，事実上，このS-OSシステムコールライブラリが，Small-CからS-OSシステムを呼び出す標準関数となります。

●今月のフリーソフト化は3本

今月，新たにフリーソフト化計画に加わったものは，

1986年11月号
第33部　Maze in Maze
1987年10月号
第50部　tiny CORE WARS
1988年7月号
第67部　マルチウィンドウドライバMW-1

以上の3本です。長嶋さん，進藤さんご協力感謝します。引き続き，フリーソフト化の募集は行っていますので，以前THE SENTINELで掲載されたプログラムを「コピーしてもいいよ」という方がいらっしゃいましたら，ぜひアンケートハガキか官製ハガキで連絡をお願いします。

●S-OSコピーサービス

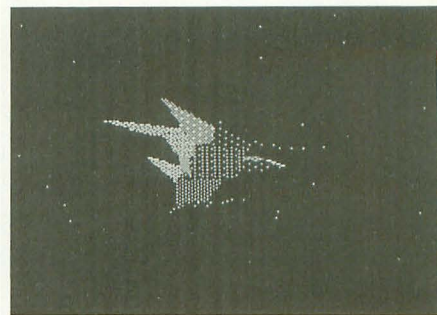
突然告知が行われたS-OSコピーサービスですが，2月25日をもって締め切らせていただきます。なお，対象となる記事は以下のとおりで，デバッグ情報はサポートしません。

・1986年2月号，X1/X1turbo，MZ-80B/2000/2200/2500，MZ-80K/C/1200/700/1500
・1986年8月号，MZ-2500用
・1987年9月号，PC-8001/8801用
・1987年10月号，X1turbo用
・1993年7月号，MSX用

官製ハガキに住所，氏名，電話番号，コピーしてもらいたい記事を明記のうえ，以下の宛先にお送りください。

Oh!X編集部
SWORDマニュアルコピー係

●POLY.KIT

POLY.KITは，以前THE SENTINELで紹介したGRAPH.LIBのポリゴン描画機能のみを奥野氏が抜き出し，強化したものです。基本的にGRAPH.LIBと同じくSLANGのライブラリ形式となっています。強化したといってもただの高速化ではなく，カラー対応で，ポリゴン描画には拡散反射，鏡面反射(ハイライトのみ)，疑似環境光までサポートするすさまじさ。さらにカラー対応にあたって，「MAGIC」自身も改造してしまったようです。まずは，サンプルデモを立ち上げてみると，どこかで見たような戦闘機が秒間3コマぐらいの速度で通り過ぎていく映像(右上の写真)と，WAVE(GRAPH.LIBのサンプル)を見せてくれました。モノクロ写真では判別がつきにくいのですが，GRAPH.LIBよりも画面の質感が上がっている感じがわかります。

仕様が複雑で初心者向きでない，改造版MAGICが現在制作されているのはX1のみ，などいろいろな問題も抱えていますが，編集部のほうでもなんとか協力して解決していきたいですね。

●グラフィックパッケージもフリーに？

S-OSアプリケーションではありませんが，高速グラフィックパッケージ「MAGIC」もコピーフリーとなりそうな気配となりました。現在，制作者の吉村氏と具体的な内容を交渉しているところです。

また，MAGICは高速なグラフィックを描画させることが主な目的ですが，それ以上にZ80マシンでのグラフィック描画の統一規格でもあるわけです。つまり，奥野氏のようにオリジナルの機能を拡張するにあたっては，慎重に再検討する必要があります。ただし，今回奥野氏が行った改造は，カラー対応するために描画モードのパラメータを拡張し，従来のMAGICとカラー対応版とを切り替えるようになっています。つまり，従来のMAGICとの完全上位互換性が保たれているのです。こういう仕様ならば，新たなカラー対応版MAGICの標準システムとなりうるかもしれません。

全機種共通
S-OS"SWORD"要

S-OS システムコール ライブラリ (SOSLIB)

Kinoshita Tatuya
木下 達也

Small-CからS-OSシステムコールを行うための関数群を紹介します。すべての関数をいっぺんに打ち込むのは，非常に大変ですので，まずは必要なものから順番に打ち込んでいきましょう。

```
 1 ;
 2 ;  soshex(hexc)
 3 ;  int hexc;
 4 ;
 5 ;  return hexadecimal number(0x0
   .  0xF).
 6 ;  if error then return -1.
 7 ;
 8
 9    INCLUDE soscall.mac
10
11 soshex::
12
13    ; A = 1st parameter low byte.
14    INC SP
15    POP BC
16    LD A,B
17    PUSH BC
18    DEC SP
19
20    CALL _HEX
21    JR C,ERR
22
23    LD L,A
```

S-OSシステムコールライブラリ(以下SOSLIBといいます)は，S-OSのサブルーチン，ワークエリアをC言語で簡単に使えるようにするためのライブラリです。

SOSLIBを使えば，従来アセンブリ言語でしか記述できなかった，S-OSに密着したきめ細かな処理を，C言語でも記述できるようになります。

ただし，SOSLIBを使うことによって，上位レベルの関数(標準ライブラリ関数など)に影響が出るかもしれません。プログラマの責任において使用してください。

使い方

ソースファイル中で，sos.hをインクルードしてください。リンク時には，soslib.LIBをリンクしてください。

例として，以下のCプログラムhellosos.Cのコンパイル手順を示します。なお，必要なファイルはすべてデフォルトデバイスにあるものとします。

```
/* hellosos.C */
#include <sos.h>
main()
  {
  sosmprnt("hello, S-OS world¥n");
  /* 文字列表示(S-OS #MPRNT) */
  }
```

S-OSのモニタのプロンプトから以下のように打ち込みます。

```
#L SC
#J3000 hellosos
#L WZD
#J3000 =hellosos
#L WLK
# J3000  /P:3000,hellosos,soslib/S,clib/S,hellosos/N:P
```

エラーがなければ，hellosos.OBJという実行ファイルができているはずです。さらに以下のように打ち込めば実行します。

```
#L hellosos.OBJ
#J3000
```

以下のように，メッセージが表示されて，プロンプトに戻れば成功です。

```
hello, S-OS world
#
```

ライブラリファイルの作り方

以下に，SOSLIBのソースファイルからライブラリファイルsoslib.LIBを作成する手順を示します。まずソースファイルsos.Cから，リロケータブルファイルsos.RELを作ります。

```
#L SC
#J3000 sos
#L WZD
#J3000 =sos
```

S-OSのサブルーチンに相当する関数のソースファイル(soscold.ASMなど)をアセンブルしてリロケータブルファイル(soscold.RELなど)を作ります。たとえば，soscold.ASMからsoscold.RELを作るには以下のようにします。

```
#L WZD
#J3000 =soscold
```

関数のソースファイル間に依存関係はないので，必要な関数だけを選んでアセンブルしてもかまいません。リロケータブルファイルをまとめて，ライブラリファイルを作ります。このとき，sos.RELは最後に指定する必要があります。

```
#L WLB
#J3000
*soscold
  ⋮
*sos
*soslib/R
```

ディスク容量不足のため，一括でライブラリファイルを作成できないときは，以下のように，分割したライブラリファイルを作成して，最後にまとめてください。

```
#L WLB
#J3000
*soscold
  ⋮
*sosinkey
*soslib1/R
#L WLB
#J3000
*sospause
  ⋮
*sospeekb
*soslib2/R
#L WLB
#J3000
*sosmon
  ⋮
*soserror
*soslib3/R
#L WLB
#J3000
*soslib1
*soslib2
*soslib3
*sos
*soslib/R
```

諸注意

本ソフトウェアは，無保証のため，ソフトを使用したために発生した事故に対しては責任を負いかねます。

本ソフトウェアは，自由に配布，改変することができます。開発には，WZDシリーズほか，多くのフリーソフトウェアを使用しました。各ソフトウェアの関係者の方々に感謝します。バグ報告などは下記宛の電子メールでお願いします。

・NIFTY-Serve：HGG03751
・S-OSnetwork大阪(S-OSユーザーズクラブ)：SOS215

S-OSシステムコールライブラリver.0.11リファレンスマニュアル

本リファレンスは，S-OSシステムコールライブラリ(以下SOSLIBといいます)の仕様についてまとめたものです。なお，S-OS "SWORD" システムのシステムコールの詳細については，Oh! MZ 1986年2月号 THE SENTINEL 第15部 ディスク対応S-OS "SWORD" などを参照するようにしてください。

外部変数

SOSLIBでは，外部変数soserrnoを定義しています。sos.hでは，以下のように宣言しています。

extern int soserrno;

SOSLIBの関数は，エラーのとき，soserrnoにS-OSエラー番号(正の整数)を設定することがあります。SOSLIBの関数は，soserrnoに0を設定することはありません。このため通常は，SOSLIBの関数を呼び出す前にsoserrnoに0を設定し，呼び出したあとにsoserrnoをチェックするという使い方をします。

定数

sos.hでは，S-OSのワークエリアのアドレス，ファイルアトリビュートの値を以下のように定義しています。通常，S-OSのワークエリアのアドレスはポインタ変数に代入して使用します。

●S-OSのワークエリアとの対応

- sosusr……#USR
- sosdvsw……#DVSW
- soslpsw……#LPSW
- sosprcnt……#PRCNT
- sosxyadr……#XYADR
- soskbfad……#KBFAD
- sosibfad……#IBFAD
- sossize……#SIZE
- sosdtadr……#DTADR
- sosexadr……#EXADR
- sosstkad……#STKAD
- sosmemax……#MEMAX
- soswksiz……#WKSIZ
- sosdirno……#DIRNO
- sosmxtrk……#MXTRK
- sosdtbuf……#DTBUF
- sosfatbf……#FATBF
- sosdirps……#DIRPS
- sosfatps……#FATPOS
- sosdsk……#DSK
- soswidth……#WIDTH
- sosmxlin……#MAXLIN

●ファイルアトリビュート

- FA_BIN……バイナリファイルのファイルアトリビュート
- FA_BAS……BASICソースファイルのファイルアトリビュート
- FA_ASC……アスキーファイルのファイルアトリビュート
- FA_RDO……ライトプロテクト(リードオンリー)ファイルのファイルアトリビュート

関数

SOSLIBでは，S-OSのサブルーチンに概当する関数を，以下のように定義しています。"S-OS:～"は，S-OSで相当するサブルーチン名です。なお，S-OSのサブルーチン[HL]，#GETPCに相当する関数はありません。

・soscold()
S-OS:#COLD
S-OSのコールドスタートを実行します

・soshot()
S-OS：#HOT
S-OSのホットスタートを実行します

・sosver()
S-OS：#VER
S-OSの機種番号とバージョンナンバーを返します。上位8ビットで機種番号を，下位8ビットでバージョンナンバーを表します

・sosprint(c)
int c;
S-OS：#PRINT
文字cを表示します

・sosprnts()
S-OS：#PRNTS
スペースを1文字表示します

・sosltnl()
S-OS：#LTNL
改行します

・sosnl()
S-OS：#NL
カーソルが行の先頭になければ改行します

・sosmsg(line)
char *line;
S-OS：#MSG
改行文字('¥n')で終わる文字列lineを表示します。改行文字は表示しません

・sosmsx(str)
char *str;
S-OS：#MSX
文字列strを表示します

・sosmprnt(str)
char *str;
S-OS：#MPRNT
文字列strを表示します

・sostab(tabpos)
int tabpos;
S-OS：#TAB
カーソルのX座標がtabposになるまで，スペースを表示します。カーソルのX座標がtabposよりも大きいときは，なにも表示しません

・soslprnt(c)
int c;
S-OS：#LPRNT
文字cをプリンタへ出力します。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します

・soslpton()
S-OS：#LPTON
画面表示と同時にプリンタへも出力をするように設定します

・soslptof()
S-OS：#LPTOF
画面表示と同時にプリンタへは出力をしないように設定します

・sosgetl(inbuf)
char *inbuf;
S-OS：#GETL
キーボードから1行入力してinbufへ格納します。入力した改行文字('¥n')はnull文字('¥0')に置き換えます。入力中にSHIFT＋BREAKが押されたなら，inbufの先頭文字をS-OS BREAK文字('¥033')にします

・sosgetky()
S-OS：#GETKY
リアルタイムキー入力をして，入力した文字を返します。キーが押されていなければ，'¥0'を返します

・sosbrkey()
S-OS：#BRKEY
BREAKキーが押されていれば0以外を返し，押されていなければ0を返します

・sosinkey()
S-OS：#INKEY
キーが押されるまで待って，キー入力します。入力した文字を返します

・sospause(adr)
char *adr;
S-OS：#PAUSE
スペースキーが押されていれば，一時停止します。SHIFT＋BREAKが押されていれば，adr番地へジャンプします

・sosbell()
S-OS：#BELL
ベル(ビープ音)を鳴らします

・sosprthx(n)
int n;
S-OS：#PRTHX
整数nを16進2桁で表示します

・sosprthl(n)
int n;
S-OS：#PRTHL
整数nを16進4桁で表示します

・sosasc(n)
int n;
S-OS：#ASC
整数nの下位4ビットの値を16進数で表す文字を返します。nが0から9なら，'0'から'9'を返します。nが10から15なら，'A'から'F'を返します

・soshex(hexc)
int hexc;
S-OS：#HEX
16進数を表す文字hexcに対応する整数を返します。hexcが16進数を表す文字なら，0x0から0xFのいずれかを返します。hexcが16進数を表す文字でなければ，－1を返します

・sos2hex(hexstr)
char ＊hexstr;
S-OS：#2HEX
2桁の16進数を表す文字列hexstrに対応する整数を返します。hexstrが2桁の16進数を表す文字列なら，0x00から0xFFのいずれかを返します。hexstrが2桁の16進数を表す文字列でなければ，－1を返します
・soshlhex(hexstr)
char ＊hexstr;
S-OS：#HLHEX
4桁の16進数を表す文字列hexstrに対応する整数を返します。hexstrが4桁の16進数を表す文字列なら，0x0000から0xFFFFのいずれかを返します。hexstrが4桁の16進数を表す文字列でなければ，－1を返して，外部変数soserrnoにS-OSエラー番号を設定します。0xFFFFと－1は同じ値なので，正確なエラーチェックをするにはsoserrnoを使う必要があります
・soswopen()
S-OS：#WOPEN
ファイル名，ファイルアトリビュート，ファイル先頭アドレス，ファイルサイズ，ファイル実行アドレスをデバイスに書き込みます
成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・soswrd()
S-OS：#WRD
データをデバイスに書き込みます。実行前にsoswopen関数を実行しておく必要があります。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosfcb()
S-OS：#FCB
インフォメーションブロックを読み込みます。BREAKキーを押すと，ディスクの先頭から読み込みを再開します。リターンキーを押すと，読み込みに失敗したとみなします。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosrdd()
S-OS：#RDD
データをデバイスから読み込みます。実行前にsosropen関数を実行しておく必要があります。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosfile(fname, fatr)
char ＊fname;
int fatr;
S-OS：#FILE
ファイル名fname，ファイルアトリビュートfatrをインフォメーションブロックに設定します。fnameによって使用するデバイスを設定します。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosfsame(fname, fatr)
char ＊fname;
int fatr;
S-OS：#FSAME
ファイル名fname，ファイルアトリビュートfatrとインフォメーションブロックのファイル名，ファイルアトリビュートを比較します。一致したなら0以外を返します。一致しないなら0を返します
・sosfprnt()
S-OS：#FPRNT
テープから読み込んだファイル名を表示します
・sospoke(wkaofst, byte)
int wkaofst;
int byte;
S-OS：#POKE
特殊ワークエリアの先頭アドレスからのオフセットwkaofstにbyteの下位8ビットを書き込みます
・sospokeb(topadr, wkaofst, nbyte)
char ＊topadr;
int wkaofst;
int nbyte;
S-OS：#POKE@
メインメモリのアドレスtopadrから，特殊ワークエリアの先頭アドレスからのオフセットwkaofstへ，バイト数nbyteのデータを転送します
・sospeek(wkaofst)
int wkaofst;
S-OS：#PEEK
特殊ワークエリアの先頭アドレスからのオフセットwkaofstに格納されている1バイトのデータを返します
・sospeekb(topadr, wkaofst, nbyte)
char ＊topadr;
int wkaofst;
int nbyte;
S-OS：#PEEK@
メインメモリのアドレスtopadrへ，特殊ワークエリアの先頭アドレスからのオフセットwkaofstから，バイト数nbyteのデータを転送します
・sosmon()
S-OS：#MON
各機種のモニタを起動します
・sosdrdsb(recno, buf, nrec)
int recno;
char ＊buf;
int nrec;
S-OS：#DRDSB
レコード番号recnoから，レコード数nrecの内容を，bufへ読み込みます。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosdwtsb(recno, buf, nrec)
int recno;
char ＊buf;
int nrec;
S-OS：#DWTSB
bufから，レコード数nrecぶんの内容を，レコード番号recnoへ書き込みます。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosdir()
S-OS：#DIR
ディレクトリの一覧を表示します
・sosropen()
S-OS：#ROPEN
インフォメーションブロックのファイル名，ファイルアトリビュートに一致するファイルをデバイスから探します。ファイルが見つかれば0を返します。ファイルが見つからなければ－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
テープから読み込んだファイル名がインフォメーションブロックのファイル名と違っていれば，1を返します
・sosset()
S-OS：#SET
インフォメーションブロックに一致するファイルにライトプロテクトを設定します。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosreset()
S-OS：#RESET
インフォメーションブロックに一致するファイルのライトプロテクトを解除します。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosname(newfn)
char ＊newfn;
S-OS：#NAME
インフォメーションブロックに一致するファイルのファイル名をnewfnに変更します
成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・soskill()
S-OS：#KILL
インフォメーションブロックに一致するファイルを削除します。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・soscsr(x, y)
int ＊x;
int ＊y;
S-OS：#CSR
カーソルのX座標を＊xに，Y座標を＊yに設定します
・sosscrn(x, y)
int x;
int y;
S-OS：#SCRN
X座標x，Y座標yの位置にある文字を返します。失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosloc(x, y)
int x;
int y;
S-OS：#LOC
カーソルをX座標x，Y座標yの位置に移動します。成功すると0を返し，失敗すると－1を返して，外部変数soserrnoにS-OSエラー番号を設定します
・sosflget()
S-OS：#FLGET
カーソル点滅1文字入力をして，入力した文字を返します。画面へのエコーバックはありません
・sosrdvsw()
S-OS：#RDVSW
デフォルトデバイス名を返します
・sossdvsw(dvname)
int dvname;
S-OS：#SDVSW
デフォルトデバイス名dvnameで，デフォルトデバイスを設定します
・sosinp(port)
int port;
S-OS：#INP
共通I/Oポートのアドレスportの1バイトの

内容を返します
- sosout(port, byte)
  int port;
  int byte;
  S-OS : #OUT
  共通I/Oポートのアドレスportに，byteの下位8ビットを設定します
- soswidch(scrmode)
  int scrmode;
  S-OS : #WIDCH
  scrmodeが40以下なら，画面モードを40文字モードに変更します。scrmodeが40より大きいなら，画面モードを80文字モードに変更します
- soserror(soseno)
  int soseno;
  S-OS : #ERROR
  S-OSエラー番号sosenoに対応するエラーメッセージを表示します

リスト1 sos.h

```
 1 /*
 2 sos.h
 3 1994-12-23 Tatsuya Kinoshita
 4 */
 5
 6
 7 #ifndef SOS_H
 8 #define SOS_H
 9
10
11 #asm
12   EXT sos
13 #endasm
14
15
16 /*
17 S-OS work area
18 */
19 #define sosusr    0x1F7E
20 #define sosdvsw   0x1F7D
21 #define soslpsw   0x1F7C
22 #define sosprcnt  0x1F7A
23 #define sosxyadr  0x1F78
24 #define soskbfad  0x1F76
25 #define sosibfad  0x1F74
26 #define sossize   0x1F72
27 #define sosdtadr  0x1F70
28 #define sosexadr  0x1F6E
29 #define sosstkad  0x1F6C
30 #define sosmemax  0x1F6A
31 #define soswksiz  0x1F68
32 #define sosdirno  0x1F67
33 #define sosmxtrk  0x1F66
34 #define sosdtbuf  0x1F64
35 #define sosfatbf  0x1F62
36 #define sosdirps  0x1F60
37 #define sosfatps  0x1F5E
38 #define sosdsk    0x1F5D
39 #define soswidth  0x1F5C
40 #define sosmxlin  0x1F5B
41
42
43 /*
44 file attribute.
45 */
46 #define FA_BIN    0x01
47 #define FA_BAS    0x02
48 #define FA_ASC    0x04
49 #define FA_HID    0x10
50 #define FA_VRF    0x20
51 #define FA_RDO    0x40
52 #define FA_DIR    0x80
53
54
55 extern int soserrno;
56
57
58 #endif /* SOS_H */
```

リスト2 soscall.MAC

```
  1 ;
  2 ; soscall.MAC
  3 ;
  4
  5 ;
  6 ; S-OS subroutines
  7 ;
  8 _COLD    EQU $1FFD
  9 _HOT     EQU $1FFA
 10 _VER     EQU $1FF7
 11 _PRINT   EQU $1FF4
 12 _PRNTS   EQU $1FF1
 13 _LTNL    EQU $1FEE
 14 _NL      EQU $1FEB
 15 _MSG     EQU $1FE8
 16 _MSX     EQU $1FE5
 17 _MPRNT   EQU $1FE2
 18 _TAB     EQU $1FDF
 19 _LPRNT   EQU $1FDC
 20 _LPTON   EQU $1FD9
 21 _LPTOF   EQU $1FD6
 22 _GETL    EQU $1FD3
 23 _GETKY   EQU $1FD0
 24 _BRKEY   EQU $1FCD
 25 _INKEY   EQU $1FCA
 26 _PAUSE   EQU $1FC7
 27 _BELL    EQU $1FC4
 28 _PRTHX   EQU $1FC1
 29 _PRTHL   EQU $1FBE
 30 _ASC     EQU $1FBB
 31 _HEX     EQU $1FB8
 32 _2HEX    EQU $1FB5
 33 _HLHEX   EQU $1FB2
 34 _WOPEN   EQU $1FAF
 35 _WRD     EQU $1FAC
 36 _FCB     EQU $1FA9
 37 _RDD     EQU $1FA6
 38 _FILE    EQU $1FA3
 39 _FSAME   EQU $1FA0
 40 _FPRNT   EQU $1F9D
 41 _POKE    EQU $1F9A
 42 _POKEB   EQU $1F97
 43 _PEEK    EQU $1F94
 44 _PEEKB   EQU $1F91
 45 _MON     EQU $1F8E
 46 _PCHL    EQU $1F81
 47 _GETPC   EQU $1F80
 48 _DRDSB   EQU $2000
 49 _DWTSB   EQU $2003
 50 _DIR     EQU $2006
 51 _ROPEN   EQU $2009
 52 _SET     EQU $200C
 53 _RESET   EQU $200F
 54 _NAME    EQU $2012
 55 _KILL    EQU $2015
 56 _CSR     EQU $2018
 57 _SCRN    EQU $201B
 58 _LOC     EQU $201E
 59 _FLGET   EQU $2021
 60 _RDVSW   EQU $2024
 61 _SDVSW   EQU $2027
 62 _INP     EQU $202A
 63 _OUT     EQU $202D
 64 _WIDCH   EQU $2030
 65 _ERROR   EQU $2033
 66 ;
 67 ; S-OS work area
 68 ;
 69 _USR     EQU $1F7E
 70 _DVSW    EQU $1F7D
 71 _LPSW    EQU $1F7C
 72 _PRCNT   EQU $1F7A
 73 _XYADR   EQU $1F78
 74 _KBFAD   EQU $1F76
 75 _IBFAD   EQU $1F74
 76 _SIZE    EQU $1F72
 77 _DTADR   EQU $1F70
 78 _EXADR   EQU $1F6E
 79 _STKAD   EQU $1F6C
 80 _MEMAX   EQU $1F6A
 81 _WKSIZ   EQU $1F68
 82 _DIRNO   EQU $1F67
 83 _MXTRK   EQU $1F66
 84 _DTBUF   EQU $1F64
 85 _FATBF   EQU $1F62
 86 _DIRPS   EQU $1F60
 87 _FATPS   EQU $1F5E
 88 _DSK     EQU $1F5D
 89 _WIDTH   EQU $1F5C
 90 _MXLIN   EQU $1F5B
 91
 92 ;
 93 ; file attribute.
 94 ;
 95 FA_BIN   EQU $01
 96 FA_BAS   EQU $02
 97 FA_ASC   EQU $04
 98 FA_HID   EQU $10
 99 FA_VRF   EQU $20
100 FA_RDO   EQU $40
101 FA_DIR   EQU $80
```

リスト3 SOSLIB全関数群

soscold.ASM

```
 1 ;
 2 ; soscold()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 soscold::
 8   JP _COLD
 9
10   END
```

soshot.ASM

```
 1 ;
 2 ; soshot()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 soshot::
 8   JP _HOT
 9
10   END
```

sosver.ASM

```
 1 ;
 2 ; sosver()
 3 ;
 4 ; return machine type, version.
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 sosver::
10   CALL _VER
11   RET
12
13   END
```

sosprint.ASM

```
 1 ;
 2 ; sosprint(c)
 3 ; int c;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosprint::
 9
10   ; A = 1st parameter low byte.
11   INC SP
12   POP BC
13   LD A,B
14   PUSH BC
15   DEC SP
16
17   JP _PRINT
18
19   END
```

sosprnts.ASM

```
 1 ;
 2 ; sosprnts()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 sosprnts::
 8   CALL _PRNTS
 9   RET
10
11   END
```

sosltnl.ASM

```
 1 ;
 2 ; sosltnl()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 sosltnl::
 8   JP _LTNL
 9
10   END
```

sosnl.ASM

```
 1 ;
 2 ; sosnl()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 sosnl::
 8   JP _NL
 9
10   END
```

sosmsg.ASM

```
 1 ;
 2 ; sosmsg(line)
 3 ; char *line;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosmsg::
 9
10   ; DE = 1st parameter.
11   POP BC
12   POP DE
13   PUSH DE
14   PUSH BC
15
16   JP _MSG
17
18   END
```

sosmsx.ASM

```
 1 ;
 2 ; sosmsx(str)
 3 ; char *str;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosmsx::
 9
10   ; DE = 1st parameter.
11   POP BC
12   POP DE
13   PUSH DE
14   PUSH BC
15
16   JP _MSX
17
18   END
```

sosmprnt.ASM

```
 1 ;
 2 ; sosmprnt(str)
 3 ; char *str;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosmprnt::
 9
10   ; DE = 1st parameter.
11   POP BC
12   POP DE
13   PUSH DE
14   PUSH BC
15
16   JP _MSX
17
18   END
```

sostab.ASM

```
 1 ;
 2 ; sostab(tabpos)
 3 ; int tabpos;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sostab::
 9
10   ; B = 1st parameter low byte.
11   INC SP
12   POP BC
13   PUSH BC
14   DEC SP
15
16   JP _TAB
17
18   END
```

soslprnt.ASM

```
 1 ;
 2 ; soslprnt(c)
 3 ; int c;
 4 ;
 5 ; if error then
 6 ;   soserrno = S-OS error number.
 7 ;   return -1.
 8 ; else
 9 ;   return 0.
10 ;
11
12   INCLUDE soscall.mac
13   EXT sos
14   EXT soserrno
15
16 soslprnt::
17
18   ; A = 1st parameter low byte.
19   INC SP
20   POP BC
21   LD A,B
22   PUSH BC
23   DEC SP
24
25   JP _LPRNT
26   JR C,ERR
27
28   LD HL,0
29   RET
30
31 ERR:
32   LD L,A
33   LD H,0
34   LD (soserrno),HL
35
36   LD HL,-1
37   RET
38
39   END
```

soslpton.ASM

```
 1 ;
 2 ; soslpton()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 soslpton::
 8   JP _LPTON
 9
10   END
```

soslptof.ASM

```
 1 ;
 2 ; soslptof()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 soslptof::
 8   JP _LPTOF
 9
10   END
```

sosgetl.ASM

```
 1 ;
 2 ; sosgetl(inbuf)
 3 ; char *inbuf;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosgetl::
 9
10   ; DE = 1st parameter.
11   POP BC
12   POP DE
13   PUSH DE
14   PUSH BC
15
16   JP _GETL
17
18   END
```

sosgetky.ASM

```
 1 ;
 2 ; sosgetky()
 3 ;
 4 ; return input character.
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 sosgetky::
10
11   CALL _GETKY
12   LD L,A
13   LD H,0
14   RET
15
16   END
```

sosbrkey.ASM

```
 1 ;
 2 ; sosbrkey()
 3 ;
 4 ; if break key on then
 5 ;   return 1.
 6 ; else
 7 ;   return 0.
 8 ;
 9
10   INCLUDE soscall.mac
11
12 sosbrkey::
13
14   CALL _BRKEY
15   JR Z,BRKON
16
17   LD HL,0
18   RET
19
20 BRKON:
21   LD L,1
22   LD H,0
23   RET
24
25   END
```

sosinkey.ASM

```
 1 ;
 2 ; sosinkey()
 3 ;
 4 ; return input character.
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 sosinkey::
10
11   CALL _INKEY
12   LD L,A
13   LD H,0
14   RET
15
16   END
```

▶ただいま試験の真っ最中です。早く春がこないかな……。　河上　博仁(18)埼玉県

sospause.ASM

```
 1 ;
 2 ; sospause(adr)
 3 ; char *adr;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sospause::
 9
10   ; HL = 1st parameter.
11   POP BC
12   POP HL
13   PUSH HL
14   PUSH BC
15
16   LD (BRADRP),HL
17
18   CALL _PAUSE
19 BRADRP:
20   DS 2
21
22   RET
23
24   END
```

sosbell.ASM

```
 1 ;
 2 ; sosbell()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 sosbell::
 8   JP _BELL
 9
10   END
```

sosprthx.ASM

```
 1 ;
 2 ; sosprthx(n)
 3 ; int n;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosprthx::
 9
10   ; A = 1st parameter low byte.
11   INC SP
12   POP BC
13   LD A,B
14   PUSH BC
15   DEC SP
16
17   JP _PRTHX
18
19   END
```

sosprthl.ASM

```
 1 ;
 2 ; sosprthl(n)
 3 ; int n;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosprthl::
 9
10   ; HL = 1st parameter.
11   POP BC
12   POP HL
13   PUSH HL
14   PUSH BC
15
16   JP _PRTHL
17
18   END
```

sosasc.ASM

```
 1 ;
 2 ; sosasc(n)
 3 ; int n;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sosasc::
 9
10   ; A = 1st parameter low byte.
11   INC SP
12   POP BC
13   LD A,B
14   PUSH BC
15   DEC SP
16   CALL _ASC
17   LD L,A
18   LD H,0
19   RET
20   END
```

soshex.ASM

```
 1 ;
 2 ; soshex(hexc)
 3 ; int hexc;
 4 ;
 5 ; return hexadecimal number(0x0
   .. 0xF).
 6 ; if error then return -1.
 7 ;
 8
 9   INCLUDE soscall.mac
10
11 soshex::
12
13   ; A = 1st parameter low byte.
14   INC SP
15   POP BC
16   LD A,B
17   PUSH BC
18   DEC SP
19
20   CALL _HEX
21   JR C,ERR
22
23   LD L,A
24   LD H,0
25   RET
26
27 ERR:
28   LD HL,-1
29   RET
30
31   END
```

sos2hex.ASM

```
 1 ;
 2 ; sos2hex(hexstr)
 3 ; char *hexstr;
 4 ;
 5 ; return hexadecimal number(0x00
   .. 0xFF).
 6 ; if error then return -1.
 7 ;
 8
 9   INCLUDE soscall.mac
10
11 sos2hex::
12
13   ; DE = 1st parameter.
14   POP BC
15   POP DE
16   PUSH DE
17   PUSH BC
18
19   CALL _2HEX
20   JR C,ERR
21
22   LD L,A
23   LD H,0
24   RET
25
26 ERR:
27   LD HL,-1
28   RET
29
30   END
```

soshlhex.ASM

```
 1 ;
 2 ; soshlhex(hexstr)
 3 ; char *hexstr;
 4 ;
 5 ; return hexadecimal number(0x0000
   .. 0xFFFF).
 6 ; if error then
 7 ;    soserrno = 14(Bad Data).
 8 ;    return -1.
 9 ;
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 soshlhex::
16
17   ; DE = 1st parameter.
18   POP BC
19   POP DE
20   PUSH DE
21   PUSH BC
22
23   CALL _HLHEX
24   RET NC
25
26   ; error.
27
28   ; soserrno = Bad Data.
29   LD HL,14
30   LD (soserrno),HL
31
32   LD HL,-1
33   RET
34
35   END
```

soswopen.ASM

```
 1 ;
 2 ; soswopen()
 3 ;
 4 ; if error then
 5 ;    soserrno = S-OS error number.
 6 ;    return -1.
 7 ; else
 8 ;    return 0.
 9 ;
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 soswopen::
16   CALL _WOPEN
17   JR C,ERR
18
19   LD HL,0
20   RET
21
22 ERR:
23   LD L,A
24   LD H,0
25   LD (soserrno),HL
26
27   LD HL,-1
28   RET
29
30   END
```

soswrd.ASM

```
 1 ;
 2 ; soswrd()
 3 ;
 4 ; if error then
 5 ;    soserrno = S-OS error number.
 6 ;    return -1.
 7 ; else
 8 ;    return 0.
 9 ;
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 soswrd::
16   CALL _WRD
17   JR C,ERR
18
19   LD HL,0
20   RET
21
22 ERR:
23   LD L,A
24   LD H,0
25   LD (soserrno),HL
26
27   LD HL,-1
28   RET
29
30   END
```

sosfcb.ASM

```
 1 ;
 2 ; sosfcb()
 3 ;
 4 ; if error then
 5 ;    soserrno = S-OS error number.
 6 ;    return -1.
 7 ; else
 8 ;    return 0.
 9 ;
```

▶ 2月号はやけに薄いような気がする。あと，どうでもいいことだけど81～96ページまで紙の質が違う。　水野　淳(19)東京都

```
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 sosfcb::
16   CALL _FCB
17   JR C,ERR
18
19   LD HL,0
20   RET
21
22 ERR:
23   LD L,A
24   LD H,0
25   LD (soserrno),HL
26
27   LD HL,-1
28   RET
29
30   END
```

sosrdd.ASM

```
 1 ;
 2 ; sosrdd()
 3 ;
 4 ; if error then
 5 ;   soserrno = S-OS error number.
 6 ;   return -1.
 7 ; else
 8 ;   return 0.
 9 ;
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 sosrdd::
16   CALL _RDD
17   JR C,ERR
18
19   LD HL,0
20   RET
21
22 ERR:
23   LD L,A
24   LD H,0
25   LD (soserrno),HL
26
27   LD HL,-1
28   RET
29
30   END
```

sosfile.ASM

```
 1 ;
 2 ; sosfile(fname, fatr)
 3 ; char *fname;
 4 ; int fatr;
 5 ;
 6 ; if error then
 7 ;   soserrno = S-OS error number.
 8 ;   return -1.
 9 ; else
10 ;   return 0.
11 ;
12
13   INCLUDE soscall.mac
14   EXT sos
15   EXT soserrno
16
17 sosfile::
18
19   ; DE = 1st parameter.
20   ; A = 2nd parameter low byte.
21   POP BC
22   POP HL
23   POP DE
24   LD A,L
25   PUSH DE
26   PUSH HL
27   PUSH BC
28
29   CALL _FILE
30   JR C,ERR
31
32   LD HL,0
33   RET
34
35 ERR:
36   LD L,A
37   LD H,0
38   LD (soserrno),HL
39
40   LD HL,-1
41   RET
42
43   END
```

sosfsame.ASM

```
 1 ;
 2 ; sosfsame(fname, fatr)
 3 ; char *fname;
 4 ; int fatr;
 5 ;
 6 ; if filename same then
 7 ;   return 1.
 8 ; else
 9 ;   return 0.
10 ;
11
12   INCLUDE soscall.mac
13
14 sosfsame::
15
16   ; DE = 1st parameter.
17   ; A = 2nd parameter low byte.
18   POP BC
19   POP HL
20   POP DE
21   LD A,L
22   PUSH DE
23   PUSH HL
24   PUSH BC
25
26   CALL _FSAME
27   JR Z,SAME
28
29   LD HL,0
30   RET
31
32 SAME:
33   LD HL,1
34   RET
35
36   END
```

sosfprnt.ASM

```
 1 ;
 2 ; sosfprnt()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 sosfprnt::
 8   JP _FPRNT
 9
10   END
```

sospoke.ASM

```
 1 ;
 2 ; sospoke(wkaofst, byte)
 3 ; int wkaofst;
 4 ; int byte;
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 sospoke::
10
11   ; HL = 1st parameter.
12   ; A = 2nd parameter low byte.
13   POP BC
14   POP DE
15   POP HL
16   LD A,E
17   PUSH HL
18   PUSH DE
19   PUSH BC
20
21   JP _POKE
22
23   END
```

sospokeb.ASM

```
 1 ;
 2 ; sospokeb(topadr, wkaofst, nbyte)
 3 ; char *topadr;
 4 ; int wkaofst;
 5 ; int nbyte;
 6 ;
 7
 8   INCLUDE soscall.mac
 9
10 sospokeb::
11
12   ; HL = 1st parameter.
13   ; DE = 2nd parameter.
14   ; BC = 3rd parameter.
15   POP BC
16   POP BC
17   POP DE
18   POP HL
19   PUSH HL
20   PUSH DE
21   PUSH BC
22   DEC SP
23   DEC SP
24
25   JP _POKEB
26
27   END
```

sospeek.ASM

```
 1 ;
 2 ; sospeek(wkaofst)
 3 ; int wkaofst;
 4 ;
 5 ; return work area byte.
 6 ;
 7
 8   INCLUDE soscall.mac
 9
10 sospeek::
11
12   ; HL = 1st parameter.
13   POP BC
14   POP HL
15   PUSH HL
16   PUSH BC
17
18   CALL _PEEK
19   LD L,A
20   LD H,0
21   RET
22
23   END
```

sospeekb.ASM

```
 1 ;
 2 ; sospeekb(topadr, wkaofst, nbyte)
 3 ; char *topadr;
 4 ; int wkaofst;
 5 ; int nbyte;
 6 ;
 7
 8   INCLUDE soscall.mac
 9
10 sospeekb::
11
12   ; HL = 1st parameter.
13   ; DE = 2nd parameter.
14   ; BC = 3rd parameter.
15   POP BC
16   POP BC
17   POP DE
18   POP HL
19   PUSH HL
20   PUSH DE
21   PUSH BC
22   DEC SP
23   DEC SP
24
25   JP _PEEKB
26
27   END
```

sosmon.ASM

```
 1 ;
 2 ; sosmon()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 sosmon::
 8   JP _MON
 9
10   END
```

sosdrdsb.ASM

```
 1 ;
 2 ; sosdrdsb(recno, buf, nrec)
 3 ; int recno;
 4 ; char *buf;
 5 ; int nrec;
 6 ;
 7 ; if error then
 8 ;   soserrno = S-OS error number.
 9 ;   return -1.
```

▶高校卒業(予定)記念にクロックアップ改造を計画中。　小野寺　健一(17)宮城県

```
10 ; else
11 ;   return 0.
12 ;
13
14   INCLUDE soscall.mac
15   EXT sos
16   EXT soserrno
17
18 sosdrdsb::
19
20   ; DE = 1st parameter.
21   ; HL = 2nd parameter.
22   ; A  = 3rd parameter low byte.
23   POP BC
24   POP BC
25   POP HL
26   POP DE
27   LD A,C
28   PUSH HL
29   PUSH DE
30   PUSH BC
31   DEC SP
32   DEC SP
33
34   CALL _DRDSB
35   JR C,ERR
36
37   LD HL,0
38   RET
39
40 ERR:
41   LD L,A
42   LD H,0
43   LD (soserrno),HL
44
45   LD HL,-1
46   RET
47
48   END
```

sosdwtsb.ASM

```
 1 ;
 2 ; sosdwtsb(recno, buf, nrec)
 3 ; int recno;
 4 ; char *buf;
 5 ; int nrec;
 6 ;
 7 ; if error then
 8 ;   soserrno = S-OS error number.
 9 ;   return -1.
10 ; else
11 ;   return 0.
12 ;
13
14   INCLUDE soscall.mac
15   EXT sos
16   EXT soserrno
17
18 sosdwtsb::
19
20   ; DE = 1st parameter.
21   ; HL = 2nd parameter.
22   ; A  = 3wt parameter low byte.
23   POP BC
24   POP BC
25   POP HL
26   POP DE
27   LD A,C
28   PUSH HL
29   PUSH DE
30   PUSH BC
31   DEC SP
32   DEC SP
33
34   CALL _DWTSB
35   JR C,ERR
36
37   LD HL,0
38   RET
39
40 ERR:
41   LD L,A
42   LD H,0
43   LD (soserrno),HL
44
45   LD HL,-1
46   RET
47
48   END
```

sosdir.ASM

```
 1 ;
 2 ; sosdir()
 3 ;
 4
 5   INCLUDE soscall.mac
 6
 7 sosdir::
 8   JP _DIR
 9
10   END
```

sosropen.ASM

```
 1 ;
 2 ; sosropen()
 3 ;
 4 ; if success then
 5 ;   return 0.
 6 ; if error then
 7 ;   soserrno = S-OS error number.
 8 ;   return -1.
 9 ; if read file is different then
10 ;   return 1.
11 ;
12
13   INCLUDE soscall.mac
14   EXT sos
15   EXT soserrno
16
17 sosropen::
18   CALL _ROPEN
19   JR C,ERR
20   JR NZ,DIF
21
22   LD HL,0
23   RET
24
25 ERR:
26   LD L,A
27   LD H,0
28   LD (soserrno),HL
29
30   LD HL,-1
31   RET
32
33 DIF:
34   LD HL,1
35   RET
36
37   END
```

sosset.ASM

```
 1 ;
 2 ; sosset()
 3 ;
 4 ; if error then
 5 ;   soserrno = S-OS error number.
 6 ;   return -1.
 7 ; else
 8 ;   return 0.
 9 ;
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 sosset::
16   CALL _SET
17   JR C,ERR
18
19   LD HL,0
20   RET
21
22 ERR:
23   LD L,A
24   LD H,0
25   LD (soserrno),HL
26
27   LD HL,-1
28   RET
29
30   END
```

sosreset.ASM

```
 1 ;
 2 ; sosreset()
 3 ;
 4 ; if error then
 5 ;   soserrno = S-OS error number.
 6 ;   return -1.
 7 ; else
 8 ;   return 0.
 9 ;
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 sosreset::
16   CALL _RESET
17   JR C,ERR
18
19   LD HL,0
20   RET
21
22 ERR:
23   LD L,A
24   LD H,0
25   LD (soserrno),HL
26
27   LD HL,-1
28   RET
29
30   END
```

sosname.ASM

```
 1 ;
 2 ; sosname(newfn)
 3 ; char *newfn;
 4 ;
 5 ; if error then
 6 ;   soserrno = S-OS error number.
 7 ;   return -1.
 8 ; else
 9 ;   return 0.
10 ;
11
12   INCLUDE soscall.mac
13   EXT sos
14   EXT soserrno
15
16 sosname::
17
18   ; DE = 1st parameter.
19   POP BC
20   POP DE
21   PUSH DE
22   PUSH BC
23
24   CALL _NAME
25   JR C,ERR
26
27   LD HL,0
28   RET
29
30 ERR:
31   LD L,A
32   LD H,0
33   LD (soserrno),HL
34
35   LD HL,-1
36   RET
37
38   END
```

soskill.ASM

```
 1 ;
 2 ; soskill()
 3 ;
 4 ; if error then
 5 ;   soserrno = S-OS error number.
 6 ;   return -1.
 7 ; else
 8 ;   return 0.
 9 ;
10
11   INCLUDE soscall.mac
12   EXT sos
13   EXT soserrno
14
15 soskill::
16   CALL _KILL
17   JR C,ERR
18
19   LD HL,0
20   RET
21
22 ERR:
23   LD L,A
24   LD H,0
25   LD (soserrno),HL
26
27   LD HL,-1
28   RET
29
30   END
```

▶友人のもっているPlayStationでやろうと思って「リッジレーサー」を買ったら，先にクリアされてしまった。おまけに黒い車に慣れたら「リッジレーサー２」がまともに走れなくなるし……。
安達　雄一(19)広島県

soscsr.ASM

```
 1 ;
 2 ; soscsr(x, y)
 3 ; int *x;
 4 ; int *y;
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 soscsr::
10   CALL _CSR
11   EX DE,HL
12
13   POP BC
14   ; HL = 2nd parameter.
15   POP HL
16   LD (HL),D
17   INC HL
18   LD (HL),0
19   ; HL = 1st parameter.
20   POP HL
21   LD (HL),E
22   INC HL
23   LD (HL),0
24
25   DEC SP
26   DEC SP
27   DEC SP
28   DEC SP
29   PUSH BC
30
31   RET
32
33   END
```

sosscrn.ASM

```
 1 ;
 2 ; sosscrn(x, y)
 3 ; int x;
 4 ; int y;
 5 ;
 6 ; return (x, y) character.
 7 ; if error then
 8 ;    soserrno = S-OS error number.
 9 ;    return -1.
10 ;
11
12   INCLUDE soscall.mac
13   EXT sos
14   EXT soserrno
15
16 sosscrn::
17
18   ; L = 1st parameter low byte.
19   ; H = 2nd parameter low byte.
20   INC SP
21   POP HL
22   POP DE
23   PUSH DE
24   PUSH HL
25   DEC SP
26   LD L,D
27
28   CALL _SCRN
29   LD L,A
30   LD H,0
31   RET NC
32
33   ; error.
34   LD (soserrno),HL
35   LD HL,-1
36   RET
37
38   END
```

sosloc.ASM

```
 1 ;
 2 ; sosloc(x, y)
 3 ; int x;
 4 ; int y;
 5 ;
 6 ; if error then
 7 ;    soserrno = S-OS error number.
 8 ;    return -1.
 9 ; else
10 ;    return 0.
11 ;
12
13   INCLUDE soscall.mac
14   EXT sos
15   EXT soserrno
16
17 sosloc::
18
19   ; L = 1st parameter low byte.
20   ; H = 2nd parameter low byte.
21   INC SP
22   POP HL
23   POP DE
24   PUSH DE
25   PUSH HL
26   DEC SP
27   LD L,D
28
29   CALL _LOC
30   JR C,ERR
31
32   LD HL,0
33   RET
34
35 ERR:
36   LD L,A
37   LD H,0
38   LD (soserrno),HL
39
40   LD HL,-1
41   RET
42
43   END
```

sosflget.ASM

```
 1 ;
 2 ; sosflget()
 3 ;
 4 ; return input character.
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 sosflget::
10   CALL _FLGET
11   LD L,A
12   LD H,0
13   RET
14
15   END
```

sosrdvsw.ASM

```
 1 ;
 2 ; sosrdvsw()
 3 ;
 4 ; return default device.
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 sosrdvsw::
10   CALL _RDVSW
11   LD L,A
12   LD H,0
13   RET
14
15   END
```

sossdvsw.ASM

```
 1 ;
 2 ; sossdvsw(dvname)
 3 ; int dvname;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 sossdvsw::
 9
10   ; A = 1st parameter low byte.
11   INC SP
12   POP DE
13   LD A,D
14   PUSH DE
15   DEC SP
16
17   JP _SDVSW
18
19   END
```

sosinp.ASM

```
 1 ;
 2 ; sosinp(port)
 3 ; int port;
 4 ;
 5 ; return common I/O port byte.
 6 ;
 7
 8   INCLUDE soscall.mac
 9
10 sosinp::
11
12   ; C = 1st parameter low byte.
13   POP DE
14   POP BC
15   PUSH BC
16   PUSH DE
17
18   CALL _INP
19   LD L,A
20   LD H,0
21   RET
22
23   END
```

sosout.ASM

```
 1 ;
 2 ; sosout(port, byte)
 3 ; int port;
 4 ; int byte;
 5 ;
 6
 7   INCLUDE soscall.mac
 8
 9 sosout::
10
11   ; C = 1st parameter low byte.
12   ; A = 2nd parameter low byte.
13   POP DE
14   POP HL
15   POP BC
16   PUSH BC
17   PUSH HL
18   PUSH DE
19   LD A,L
20
21   JP _OUT
22
23   END
```

soswidch.ASM

```
 1 ;
 2 ; soswidch(scrmode)
 3 ; int scrmode;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 soswidch::
 9
10   ; A = 1st parameter low byte.
11   INC SP
12   POP DE
13   LD A,D
14   PUSH DE
15   DEC SP
16
17   JP _WIDCH
18
19   END
```

soserror.ASM

```
 1 ;
 2 ; soserror(soseno)
 3 ; int soseno;
 4 ;
 5
 6   INCLUDE soscall.mac
 7
 8 soserror::
 9
10   ; A = 1st parameter low byte.
11   INC SP
12   POP DE
13   LD A,D
14   PUSH DE
15   DEC SP
16
17   JP _ERROR
18
19   END
```

sos.C

```
 1 /*
 2 sos.C
 3 */
 4
 5 #asm
 6 sos::
 7 #endasm
 8
 9 int soserrno = 0;
10
```

こちらシステムX探偵事務所

統計資料を使う

Shibata Atushi 柴田　淳

今回は社会科学系のシミュレーションを作成するうえで非常に重要なヒントを与えてくれる統計資料の活用方法を紹介します。サンプルプログラムでは限られた条件で商業立地をシミュレートしています。

FILE-XX

illustration：T.Takahashi

柴田淳（以下Ats）：雑学クイーズ！

マスター（以下M）：おっ，今回はいきなりテンションを上げてきましたね。

Ats：さあ，問題はアシスタントの春香ちゃんからどうぞ。

琴張春香（以下春）：ちょっと，なんでわたしがアシスタント？　だいたい問題なんてわたし知らないし……，え，ああ，この紙を読めばいいのね。ええと，「一般に政府の報告書は白書と呼ばれますが，なぜでしょう」

琴張護（以下護）：2人が司会とアシスタントということなら，私とマスターが解答者ということですか。

M：うーん，白書の命名の由来ですか？　たしか本の表紙が白かったからそう呼ばれるようになったんですよね。

護：正確には，イギリスの外交報告書の表紙が白く，ホワイト・ペーパーと呼ばれていたために，政府の報告書を白書と呼ぶのが通例になったのです。ほかにもイギリスの議会報告書の表紙が青いことから青書という呼ばれ方をする政府刊行物があったり，またほかの欧州各国では，ドイツの赤，フランスの黄色など，表紙の色が通称となっている例がいくつもあるある……。

Ats：なにもそんなに詳しく答えなくてもいいですよ。

護：やはりこういうことは完璧でないと。

M：で，白書がどうしたんですか？

Ats：いや，政府の報告書というとお固いイメージがありますけど，これをよく読むとなかなか役に立ちそうなことがいろいろ載っているんです。

春：役に立ちそうって，社会科学系のシミュレーションゲームを作るのに？

護：なるほど。実際の現象をシミュレートしようとするときに，国内外の情勢分析や統計資料が載っている報告書というのは，なかなか使い道が多そうです。

Ats：そうなんですよ。それに，将来の予測などに関しては，実際に予測や分析に用いた計算式や，その根拠なんかも付記されているんです。つまり，この計算式を「計算モデル」として使えば，それなりにシミュレーションが成り立ってしまうんですよ。

M：へえ，そうなんですか。じゃあ，たとえば，どんなのが載ってるんですか？

Ats：平成6年度版の『経済白書』を見ると，円高による輸入物価の下落が国内の物価にどういう影響を与えるか，という推計が載っているんです。で，最終的に消費者物価の要因となる，卸売物価を推計するわけですが，それは次のような式で導かれるんだそうです。

国内卸売物価指数＝α×輸入物価指数＋β×稼働率指数＋γ×単位稼働コスト＋(－0.55389)＋DM　……………(式1)

ただし，$\alpha=0.12165$，$\beta=0.42497$，$\gamma=0.44216$，DMはダミー変数

春：あの，これってバリバリ経済学じゃないの？

M：そうですよね。輸入卸売物価指数だとか単位稼働コストだとか，見ているだけでくらくらしそうだ。

Ats：用語の難しさはあるけど，ここに書いてある数値はそこらへんの統計資料からもってくればいいし，それに，式の形はたんなる1次式だし。

春：でも，いきなりこんな式を目の前に突き出されて「どうだ！」みたいにされても腰が引けちゃう。

重回帰分析による予測

Ats：物価指数のようなものは値を決定する要因が複雑なので，もう少し簡単な要素で決まる統計値を例にとりましょうか。

春：簡単な要素で決まる統計値？

Ats：たとえば，ある国の人口と発電量には，比例関係が成り立つことが知られています。縦軸に日本の発電量，横軸に人口をとってグラフにプロットしてみると，点がほぼ一直線上に並ぶでしょ。

M：本当ですね。比例関係ってことは，人口と発電量の関係は1次関数で表すことができる，というわけだ。

護：ただし，1次関数の傾きと切片を決める必要があります。

M：傾きや切片なんかは，実際に測定した値があるわけだから，それを連立方程式かなにかに代入して，出てきた値を平均すれば決まるんじゃないですか？

Ats：いや，それではさすがに大雑把すぎますよ。

春：どうして大雑把すぎるの？

Ats：人口と発電量の間に比例関係が成り立つといっても，そこにはかならず誤差が伴うはずですよね。

護：つまり，実測値から直接傾きなどを導き出した場合，誤差が紛れ込んでしまい具合が悪いのです。

Ats：複数の標本値から，そこに誤差のあることを考慮しつつ未知のパラメータを導き出すための方法はありますが，これには統計学の知識が必要になります。

春：さすがにそこまで勉強するのは面倒ね。

Ats：まあいずれにしろ，どこかの誰かが

すでに計算した値があるわけですから，それを使えばいいんですよ。

M：で，関数の傾きと切片はどうなるんですか？

Ats：過去20年くらいの実測値を参考にすると，日本の発電量と人口の間には，

発電量≒−14126.4＋1.6184×人口
（ただし人口は万人）

という関係があるようです。この式が，図1の点線で描いてある直線に相当します。なお，このような関数式は「単純回帰分析モデル」と呼ばれています。

護：図を見るとグラフに打った点と直線の間に微妙なズレが見えますが，これは誤差，というわけですね。

Ats：誤差というのは偶然の要素によって決まるわけですから，完全に予測するのは不可能です。またGDP(国内総生産)のように僅かな予測誤差が大きな影響を与える予測の場合はかなりの厳密性が求められると思いますが，経済予測などに使われる手法をゲームに使おうというのだから，それほど正確な予測は必要ないはずです。

春：ところで，少し気になったんだけど，図1のグラフに，どうして1984年の値までしかないの？

Ats：率直にいうと，1985年以降の値をグラフ化すると，人口と発電量の関係が直線にならないから。実際にグラフを見てもらったほうが早いかな（図2）。

M：本当ですね。1985年と1988年で，ガクッと発電量が上がってますね。

護：バブル経済の影響でしょうか。

春：それにしても，年間で発電量が2倍以上になるっていうのは凄まじいわね。

M：こんなふうに統計値が乱高下する場合でも，きちんとした予測値を導き出せるような方法はないんですか？

Ats：ひとつには，発電量を導き出す関数に，人口以外の要素を加える，という方法が考えられます。つまり発電量を決めるのに，電気を使う人間の数のほかに，発電量に関して支配的な数量を変数として使うわけです。

護：人口以外に発電量を決める要素といえば，GDPなどはどうでしょうか。

Ats：GDPだと値が大きすぎるので，実質経済成長率を使いましょうか。すると，

発電量≒−14898.3＋1.6782×人口＋17.2814×実質経済成長率
（ただし人口は万人）

という式が導かれるそうです。このように変数が2つ以上ある予測モデルを「重回帰分析モデル」と呼びます。

M：この式を使って1988年の発電量を計算してみるとどうなるんですか？

護：約5810 GWHとなります。どうやら実際の値とは，まだかなりの開きがありますね。

Ats：あれ？　おかしいなあ。実質経済成長率を含めれば，かなり正確な予測ができるはずなんだけどなあ。

M：あ，別の統計表を見ると，1987〜88年にかけては，先進国で軒並み電力消費量(≒発電量）が2倍以上になってますよ（表1）。

Ats：すると1988年というのは，二酸化炭素排出規制かなにかの理由で，全世界的にエネルギー需要が電力へとシフトしていく転換期，ということになりそうですね。

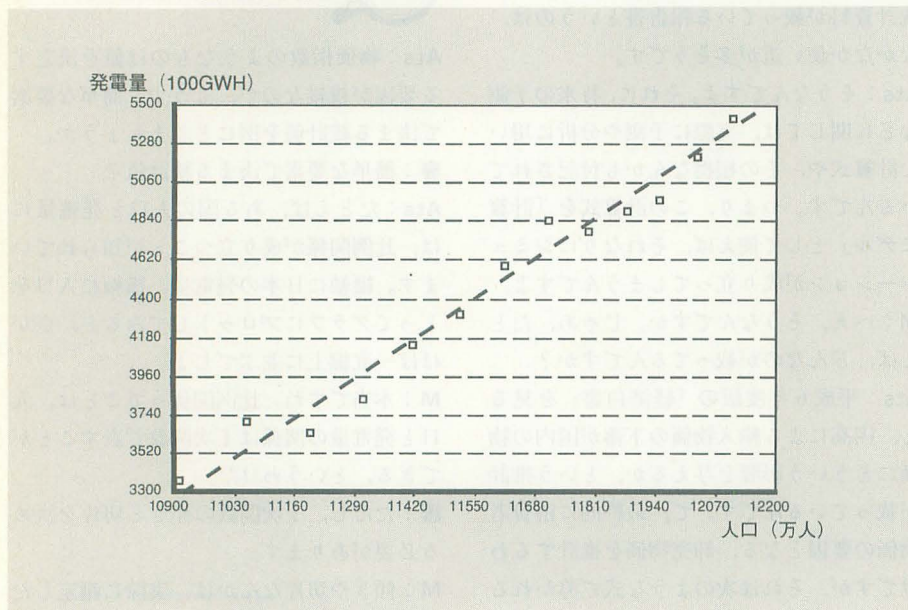
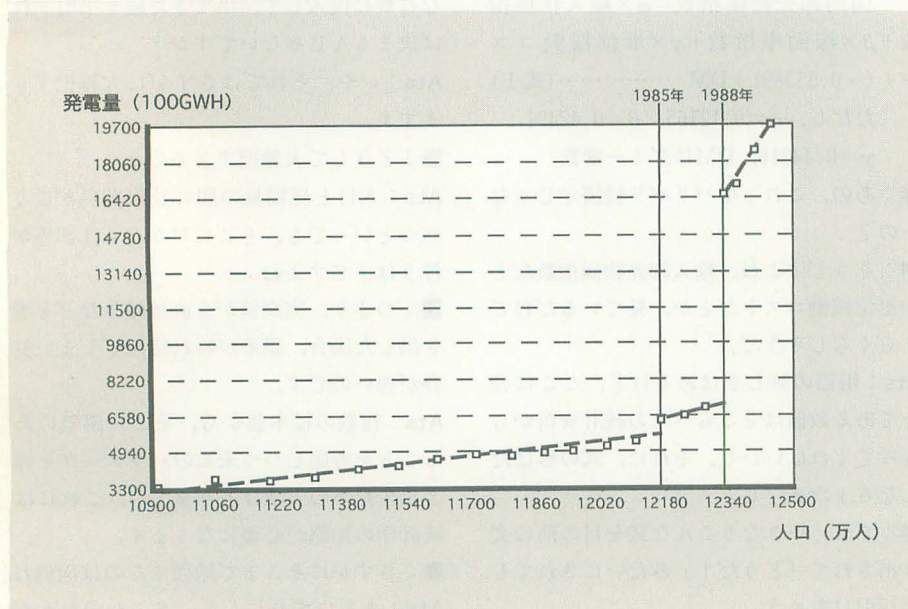
護：このように政策や技術革新などの特別な要因が統計値に影響を与える場合は，また別な手法を使う必要がありそうです。

図1　発電量と人口の関係(1972年〜1984年)

図2　発電量と人口の関係(1972年〜1991年)

表1　各国の電力消費量

	1986	1987	1988
日本	31,415	33,162	79,815
アメリカ合衆国	88,908	89,241	240,195
フランス	37,186	41,574	112,034
ドイツ	16,736	18,325	56,159
南米諸国(参考)	36,305	37,506	42,670

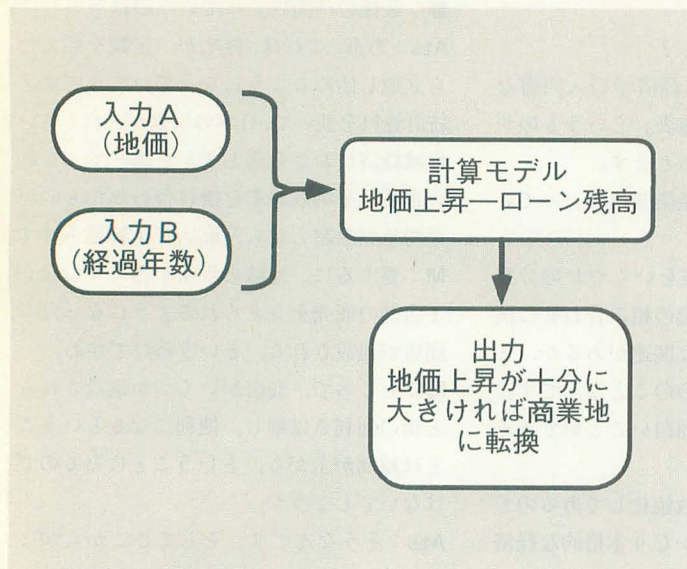
（単位は石炭換算千トン）

Ats：こういう場合は，「ダミー変数」を使うのが一般的なようです。つまり，条件をつけてある一定時期以降は予測値を底上げしたり，変数に掛け合わせる係数をある期間だけ別のものを使ったりするわけです。
M：なるほど。こうしてみると，さっきの式1もそれほど複雑な式には見えなくなってきますね。
Ats：でしょう？　本当に難しい，関数の係数はすでに出ているわけだから，あとはそれをパクればいいだけの話なんですよ。

政府刊行物をどう読むか

Ats：いろいろ本を読んでみると，どうやら経済の統計分析を行うためのソフトウェアが売られているようです。統計値自体は『白書』や『国勢図会』などを買ってくればわかります。あとはそれなりに統計学の知識があれば，自分でシミュレーションモデルを計算することができるわけです。
M：あれ，官庁の役人が計算したものがあるのだから，それを使えばいいと，さっきいってませんでしたか。
Ats：そういえばそうですね。
春：そうはいっても，始めのほうで例に出た式1なんかは1次式で簡単そうだけど，本当はこういうのは珍しくて，ほとんどが難しい計算が必要な関数ばかりなんじゃないのかしら。
Ats：ところが，経済統計などで用いられる予測モデルは1次式の「重回帰モデル」と呼ばれるものがほとんどだそうです。たまに1次式でない「非直線回帰モデル」もあるようですが，ほとんどは2次式止まりで，3次，4次や三角関数などが用いられるのはかなり特別な場合だけみたいです。
M：なるほどね。
護：しかし，政府刊行物などに載っていないようなモデルもあるのではないでしょうか。
Ats：そういう場合はマクロ経済学なんかを扱った専門書を読まなければならないわけですが，そんなに難しい計算モデルは使わないと思います。
春：でも，基本的な用語などに関しての知識もあったほうがいいだろうから，簡単な入門書くらいは読んでおいたほうがいいかもしれないわね。
Ats：それは当然ですよ。それにその手の本を読むと，ほかにもいろいろ面白い情報が手に入りますし。
M：面白い情報というと？
Ats：たとえば，マクロ経済学の入門書なんかを読むと「産業関連表」というものが紹介されていることがあります。
春：なんなの，その「産業関連表」っていうのは。
Ats：たくさんある産業をいくつかの分野に分けて，すべての産業の組み合わせに関してどれくらい経済的な関連があるか，ということを表にしたもののことなんです。これを使うだけでも，面白いことができそうでしょう？
護：産業分野の関連が数値化してあるのですから，参考にすればかなり本格的な経済シミュレーションができそうですね。
Ats：なにか知りたいことがあるとして，その情報がどこにあるかを把握しておく，というのはけっこう重要なことだと思います。で，常に情報源のありかを把握しておくためには，やっぱりいろいろな分野に目を光らせておくことが必要になってくるような気がします。
M：でも，すべての分野に精通するのは無理だろうから，広く浅く，っていうふうになるんでしょうけどね。

商業立地のシミュレーション

Ats：さて，せっかくですから政府の刊行物に載っている統計値や計算モデルなどを使って，簡単なシミュレーションを行ってみましょう。まず，幹線道路の引かれた正方形の街を想定して，ここに住宅が建っていくとします。この住宅はなるべく便利な場所に建つようにします。
護：「便利さ」はどのように表現されるのでしょうか。
Ats：次のような場所を便利，ということにします。
1)　幹線道路に近い
2)　住宅が密集している
3)　商店が近い
春：最後の「商店」っていうのは？
Ats：ああ，これは，住宅が一定数を超えたら立地し始めるようになっているんです。統計資料を使って，日本の家庭がどれくらい地域経済にお金を落とすかを調べて，これに企業の平均収益率を掛け合わせたものが平均年収を超えたら，商店が1軒建ちます。
M：要するに，地域の世帯が落とすお金が1店舗の商売を支えられるようになったら，商店が建設される，というわけですね。
護：ところで，商店がいくつか建設されると街の便利さは増し，便利になるということは地価が上がる，ということになるのではないでしょうか。
Ats：そうなんです。そしてここからがミソなんですけど，ある程度地価が上がると，住宅地が商店に置き換わるようになっているんです。
春：それは，家の値段が上がるから売っちゃうってこと？
Ats：ええと，実際はもう少し複雑です。まず，住宅はすべてローンで建てられるということにしてあるんですが，地価が上がって，家を売ればローンの残りを払ってさらにほかの同等の家が買えるとなったら，誰でも家を売るはずだ，という仮定をしました。これは，平成6年版の経済白書に載っていた，「一戸建てを買ったときの家計の超過収益率」というのをヒントに思いついたんです。
M：なになに，「住宅を購入するかどうかの意思決定に際してもうひとつの重要なポイ

図3 サンプルプログラムの方法論

ントは，住宅を購入する場合に得られる便益と費用の相対的な大きさである」だって？　つまりどういうことなんですか。

Ats：賃貸住宅に住んでいる人は毎月家賃を払っているわけですが，一戸建てを買おうとすると，普通は家賃以上のローンを払わなければならない。

護：賃貸に住み続けるとすると，家賃と住宅ローンの差額は貯蓄に回すことができ，金利が生まれます。

Ats：そのかわり，家を買った場合は地価の上昇で家自体の価値は上がるわけです。つまり，とある世帯が持家を買おうとするとき，

　超過収益率＝地価の上昇－ローン金利－ローンと家賃の差分にかかる金利

という値が大きいほうが，一戸建てを買う人が増える，というわけです。今回のシミュレーションでは，便宜上金利を一定として，持家購入時からの経過年数でローン残高を計算して，地価の上昇がローン残高を大きく上回っていれば住宅地が商業地に変わります。

春：どうして住宅地が商業地に変わるの？

Ats：ある場所の地価が一定以上に上がると，持家よりも収益率の高い商店やオフィスに転用しなければ採算が合わないからです。実際にプログラムを走らせてみるとわかると思うんですけど。

M：画面の白い点が幹線道路で，緑が住宅，青が商用地ですね。

Ats：世帯数が一定以上になると，まばらな分布の商店のほかに，商店の集まったブロックがあらわれるでしょう。これが住宅地が商業地に転換された場所なんです。

M：ところでこのシミュレーション，最初のあたりはそれらしい動きをするけど，街が大きくなると商店のブロックが偏ってしまうみたいですね。

Ats：今回のプログラムでは，地価をマップで管理しています。道路の側はあらかじめ地価が高めになっています。

護：住宅が道に沿って広がっていくのはそのためですね。

Ats：当然，住宅や商店が建つとその周りの地価が上がります。で，この地価の情報だけをもとに計算を行っているんですが，やはり情報不足なんでしょうね。現実には，都市計画や容積率などの行政の規制が街の形成に影響を及ぼすでしょうし。

護：また，地価の評価方法や，金利変動によっても街の形態は影響されるでしょう。

Ats：図3を見てください。今回のプログラムではこんな仕組みで住宅を商業地に転換しています。入力は，当然マップから得られます。複雑でそれらしく見えるシミュレーションを作ろうと思ったら，まず入力をもっと増やさなければならないでしょう。

M：でも，別に経済予測をするわけではないのですから，あまりリアルでもかえってつまらなくなるような気がしますけど。

Ats：そうなんですよね。ゲームとしてのバランスをとるためには，パラメータを微調整すればいいと思うんですが。

護：今回だったら金利や地価の評価方法などに調整の余地があるようです。

Ats：これぱっかりは実際に試してみるしかないでしょうから，適当な計算モデルが見つかったとしても，この「バランス」をとるのが結構難しいような気がします。

M：でも「シムシティー」なんかを見ると，区画の利用をプレイヤーが制限することで街を育てていくじゃないですか。つまり，計算だけではどうしてもバラついてしまう部分は，人間側から指定させる，というのが常套手段のような気がしますけど。

Ats：いずれにしろ，計算モデルに用いる入力をどのようにして増やしていくか，というのが，今後の課題かもしれません。

（つづく）

リスト1

```
 1:  /*   こちらシステムX探偵事務所*/
 2:  /*       1995年3月号サンプルプログラム*/
 3:  /*       --   任意のキーで終了 --*/
 4: #include        "stdio.h"
 5: #include        "basic0.h"
 6: #include        "BASIC.h"
 7: #include        "graph.h"
 8:
 9: #define House   2
10: #define Shop    3
11: #define Road    4
12: #define Edge    255
13: #define Sadd    10
14: #define Radd    30
15:
16: int     valu[129][129],atr[129][129],cmb[129][129];
17: int     bval[129][129],byear[129][129];
18: int     values=  0,houses = 0,shops = 0,mv = 0;
19: int     mv_x,mv_y,roads = 6 /* 道の本数(偶数) */;
20: int     year=  0,sval = 0;
21: double  cp=  0.0040410752,vex = 7;
22:
23: int     build();
24: int     refresh();
25: int     build_house();
26: int     build_shop();
27: int     build_a_shop();
28: int     add_value();
29: int     atr_to();
30: int     sell();
31: int     find_mv();
32: int     find_mv2();
33: int     draw_cell();
34: int     init_screen();
35: int     init_variables();
36: int     gen_road();
37:
38: void    main(b_argc,b_argv)
39: int     b_argc;
40: char    *b_argv[];
41: {
42:  init_screen();
43:  init_variables();
44:  do
45:  {
46:    build();
47:  }
48:  while( !kbhit() );
49:  exit(0);
50: }
```

```
 51:
 52: int     build()
 53: {
 54:   if( houses*cp - shops >=1 )
 55:   {
 56:    build_shop() ;
 57:   } else {
 58:    build_house() ;
 59:   }
 60:   year=  year + 1;
 61: }
 62:
 63: int     build_house()
 64: {
 65: int     i,j,k,x,y,v;
 66:   v = values;
 67:   if( v > 10 ){ v = 10; }
 68:   for(k= 0 ;k<= v;k++)
 69:   {
 70:    if( sval > 0 ){ sval --; }
 71:    atr_to(mv_x,mv_y,House);
 72:    x=  mv_x ; y=  mv_y;
 73:    for(i= 0 ;i<= 3;i++)
 74:    {
 75:     for(j= 0 ;j<= 3;j++)
 76:     {
 77:      add_value(x+i,y+j,1);
 78:      add_value(x+i,y-j,1);
 79:      add_value(x-i,y+j,1);
 80:      add_value(x-i,y-j,1);
 81:     }
 82:    }
 83:    houses =  houses + 1 ;
 84:    bval[x][y]=  valu[x][y] + 1000/vex;
 85:    byear[x][y] = year;
 86:    if( atr[mv_x][mv_y] != 0 ) find_mv(mv_x,mv_y);
 87:   }
 88: }
 89:
 90: int     build_shop()
 91: {
 92:   build_a_shop(mv_x,mv_y);
 93:   shops=  shops + 1 ;
 94: }
 95:
 96: int     build_a_shop(int x,int y)
 97: {
 98: int     i,j,k,v;
 99:   v = values;
100:   if( v > 50 ){ v = v*2/3; }
101:   if( v > 100 ){ v = 100; }
102:   if( sval > v ){ return(0); }
103:   atr_to(x,y,Shop);
104:   sval++;
105:   for( k = 1; k <= 8; k++ );
106:   {
107:    for(i= 0 ;i<= k;i++)
108:    {
109:     for(j= 0 ;j<= k;j++)
110:     {
111:      if( i*i+j*j < 64 )
112:      {
113:       add_value(x+i,y+j,Sadd);
114:       add_value(x+i,y-j,Sadd);
115:       add_value(x-i,y+j,Sadd);
116:       add_value(x-i,y-j,Sadd);
117:      }
118:     }
119:    }
120:   } ; values=  values + 1;
121:   if( atr[mv_x][mv_y] != 0 ){find_mv(mv_x,mv_y);}
122: }
123:
124: int     add_value(int x,int y,int v)
125: {
126: int     i,j;
127:   if( kbhit() ) exit(0);
128:   if( x > 0 & y > 0 & atr[x][y] < Shop )
129:   {
130:    if( x < 127 & y < 127 )
131:    {
132:     valu[x][y]=  valu[x][y] + v;
133:   if( valu[x][y] >= mv & atr[x][y] == 0 )
134:   {
135:    mv=  valu[x][y] ; mv_x=  x ; mv_y=  y ;
136:   }
137:   if( v == Sadd & atr[x][y] == House ){
138:    if( sell(x,y) == 1 ){
139:     build_a_shop(x,y) ;} }
140:    }
141:   }
142: }
143:
144: int     atr_to(int x,int y,int a)
145: {
146:   if( x > 0 & y > 0 ){
147:    if( x < 127 & y < 127 ){
148:    atr[x][y]=  a ; draw_cell(x,y) ;} }
149: }
150:
151: int     sell(int x,int y)
152: {
153: double  dval,left,ye;
154:   if( atr[x][y] != House || bval[x][y] == 0 )
155:   { return(0); }
156:   dval=  (valu[x][y] - bval[x][y] + 1000/vex)*vex;
157:   ye=  year - byear[x][y];
158:   if( ye > 240 ){ye=  240;}
159:   left=  ((bval[x][y]*vex+2000)*4.5)*(1.0-ye/240.0);
160:   if( left < dval )
161:   {
162:    return(1) ;
163:   } else {
164:    return(0);
165:   }
166: }
167:
168: int     find_mv(int x,int y)
169: {
170: int     i,j,k = 0,xx = -1,yy;
171:   for(i= x-5 ;i<= x+5;i++)
172:   {
173:    for(j= y-5 ;j<= y+5;j++)
174:    {
175:     if( i > 0 & y > 0 ){
176:      if( i < 127 & j < 127 ){
177:       if( valu[i][j] > k & atr[i][j] == 0 ) {
178:        k=  valu[i][j] ; xx=  i ; yy=  j ;} } }
179:    }
180:   }
181:   if( xx != -1 ){mv_x=  xx ; mv_y=  yy ; mv=  k;}
182:   if( xx == -1 ){find_mv2();}
183: }
184:
185: int     find_mv2(int x,int y)
186: {
187: int     i,j,k = 0,xx = -1,yy;
188:   for(i= 1 ;i<= 126;i++)
189:   {
190:    for(j= 1 ;j<= 126;j++)
191:    {
192:     if( i > 0 & y > 0 ){
193:      if( i < 127 & j < 127 ){
194:       if( valu[i][j] > k & atr[i][j] == 0 ) {
195:        k=  valu[i][j] ; xx=  i ; yy=  j ;} } }
196:    }
197:   }
198:   if( xx != -1 ){mv_x=  xx ; mv_y=  yy ; mv=  k;}
199: }
200:
201: int     draw_cell(int x,int y)
202: {
203:   if( atr[x][y] < 16 ){
204:   fill(x*4,y*4,x*4+3,y*4+3,atr[x][y]) ;}
205: }
206:
207: int     init_screen()
208: {
209:   screen( 2,0,1,1 ); console('NASI','NASI',0);
210:   palet(0,0 );
211:   palet(1,rgb(31,6,0) );
212:   palet(2,rgb(7,18,5) );
213:   palet(3,rgb(0,10,25) );
214:   palet(4,rgb(31,31,31) );
215: }
216:
217: int     init_variables()
218: {
219: int     i,j;
220:   for(i= 0 ;i<= 127;i++)
221:   {
222:    for( j = 0; j <= 127; j++ )
223:    {
224:     atr[i][j] = 0; valu[i][j] = 0;
225:     bval[i][j] = 0; byear[i][j] = 0;
226:    }
227:   }
228:   for(i= 0 ;i<= 127;i++)
229:   {
230:    atr[i][0]=  Edge ; atr[i][127]=  Edge;
231:    atr[0][i]=  Edge ; atr[127][i]=  Edge;
232:   }
233:   gen_road(roads);
234: }
235:
236: int     gen_road(int roads)
237: {
238: int     i,j,k,l;
239:   k=  b_int(rnd()*(128/(roads/2)))+10;
240:   for(i= 1 ;i<= roads/2;i++)
241:   {
242:    for(j= 1 ;j<= 126;j++)
243:    {
244:     for(l= -3 ;l<= 3;l++)
245:     {
246:      add_value(j,k+l,Radd-abs(l));
247:      if( l == 0 ){
248:       atr_to(j,k,Road) ;}
249:     }
250:    }
251:    k=  k + b_int(rnd()*(128/(roads/2)))+10;
252:   }
253:   k=  b_int(rnd()*(128/(roads/2)))+10;
254:   for(i= 1 ;i<= roads/2;i++)
255:   {
256:    for(j= 1 ;j<= 126;j++)
257:    {
258:     for(l= -3 ;l<= 3;l++)
259:     {
260:      if( l == 0 ) {
261:       atr_to(k,j,Road) ;
262:      } else {
263:       add_value(k+l,j,Radd-abs(l)) ;
264:      }
265:     }
266:    }
267:   k=  k + b_int(rnd()*(128/(roads/2)))+10;
268:   }
269: }
```

ピコピコエンジン活用講座(その1)

ピコピコエンジンの基礎

Ishida Norihito 石田 伯仁

SX-BASICの苦手分野をサポートしSX-WINDOW上で手軽なゲーム作成を可能にするピコピコエンジン。その威力はPushBon!.SXBですでにご存じでしょう。今回から短期連載でその使い方を解説します。まずは基本的な使い方から見ていきましょう。

リスト1，2の実行結果

もともとピコピコエンジンは，石上さんに「SX-BASICでこういうことをできるようにして！」とお願いするために作った，いわばデモソフトのようなものです。それがなんの因果か，私がピコピコエンジンの解説をすることになりました。世の中なにが起こるかわかりませんね。

なにができるのか

SX-BASICでリアルタイムゲームを作ろうとすると，ハタと困ってしまいます。キャラクタはビットマップにして，移動はmoveメソッドを使って……背景の迷路とかはどうすればいいんだろ？　パターンのアニメーションとか重ね合わせとかはちょっと難しそう。気軽にゲームでも作ってみようか，とはなかなかいきません。

もともとSX-BASICはSX-WINDOWのプログラムを簡単に作れるようにと設計されたのでしょうから，あまりゲーム作りに向いていないのは仕方ありません。しかし，仮にもBASICの名を冠する以上，ピコピコゲームは避けて通れない道です(偏見度80%)。

ピコピコエンジンはそういうSX-BASICの不得意なところ，つまりグラフィック表示を外部タスクで補うものです(SX-BASICで特殊ビットマップがサポートされるまでの命ですが……)。

なんとなく使ってみる

なにはともあれ使ってみることにしましょう。リスト1のWINOPEN.SXBを見てください。ピコピコエンジンを起動してウィンドウを開くだけのプログラムです。

ピコピコエンジンの起動にはfock()関数を使い，コマンドラインにはSX-BASICのタスクIDを指定します。まあ，おまじないだと思って例と同じに書くようにしてください。またfock()関数の返り値(ピコピコエンジンのタスクID)はピコピコエンジンに命令を送りつけるときに必要ですので，変数にしまっておきます。

命令を送りつけるにはsendmes()関数を使います。sendmes()関数の引数は，送信相手のタスクIDと送信する文字列です。ピコピコエンジンに送る文字列は，だいたい，

"命令 引数，引数，……"

のようなかたちになっています。命令(大文字のみ)のあとにスペースがあって，引数が必要なだけカンマで区切られて並んでいます。引数は，数値の場合は10進数(の文字列)で書き，文字列の場合はそのまま書きます。

ウィンドウを開くにはWINOPEN命令を送ります。

●WINOPEN タイトル，横幅，縦幅，タイプ

指定の大きさ，タイトルのウィンドウを開きます。タイプは以下のとおりです。

0　普通のウィンドウ
1　タイトル部の広いウィンドウ

さて，これでウィンドウを開くことができました。……が，なにかめちゃくちゃな模様が表示されてるかもしれません。ピコピコエンジンは「どーせデモだから」と結構いいかげんに作ってあったりするのです(もう直しましたが)。とりあえず画面クリアでもすることにしましょう。

●CLS 背景色

指定の色を背景色に設定して，ウィンドウ内を塗りつぶします。色は以下のとおり。

8　白
9　薄灰
10　濃灰
11　黒
12　黄
13　赤
14　緑
15　青

ここで「じゃあMZ-700みたいに背景を青にしてみるか……あら真っ黒になっちゃったよ～」という人がいるかもしれません。実は最初は白黒4階調しか使えないようになっているのでした。

●APAGE アクセスページ

アクセスページを設定します。使用する色の宣言のようなものです。

APAGE 3　で白黒4階調のみ
APAGE 7　で4階調＋4色

が使えるようになります。

ウィンドウだけ開いても面白くないので，文字でも書いてみます。文字関連の命令は次のようなものがあります。

リスト1　WINOPEN.SXB

```
 1: /*
 2: /*        ウィンドウを開くだけ
 3: /*
 4: int t
 5:
 6: t=fock("ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ.X "+str$(taskid))
 7:
 8: sendmes(t,"WINOPEN 只の窓,240,200,1")
 9:
10: end
```

●COLOR 描画色[, 背景色]

描画色と背景色を設定します。背景色は省略可能です。

●LOCATE x, y

ペン位置(描画開始位置)を移動します。

x, yはドット単位で指定します。

●PRINT 文字列

現在のペン位置から文字列を書きます。

●FONTKIND フォント種類

描画する文字のフォントを設定します。フォント種類は,

0 ROM12ドット
1 ROM16ドット
2 ROM24ドット

が指定できます。

●FONTSIZE 横幅, 縦幅

描画する文字の大きさを設定します。全角文字の大きさなので注意してください。

●FONTFACE [BIOUS]

文字の装飾を設定します。装飾の指定は,

B 強調
I 斜体(イタリック)
O 白抜き
U 下線つき
S 影文字

です。強調&下線にしたい場合は,

FONTFACE BU

とします。装飾なしのときは,

FONTFACE

とだけ書いてください。

このへんの機能を使ったサンプルがリスト2のPRINT.SXBです。

パターンを描いてみる

文字を書くのにも早々と飽きてしまったところで, 今度はパターンの表示を行ってみます。まずは下準備として, リソースファイル(*.LB)をダブルクリックするとパターン一覧.Xが起動するようにしておきます。アイコンメンテで*.LBの設定を,

実行ファイル　　パターン一覧.X
実行オプション　-o%

とし, 再起動してアイコンデータをファイルに保存します。

ピコピコエンジンでは, ひとつのウィンドウに表示するパターンはPAT4タイプのリソースとしてひとつのリソースファイルにまとめて入れておくことになっています。このリソースファイルをなにもないところから作るのは大変面倒くさいので, そのへんからコピーしてきて中身のパターンを全部削除してしまうのが楽ちんです。

XL/Imageお試し版+αに入っていたスクロール.LBなんかは, PAT4タイプのパターンしか入ってないのでちょうどよいでしょう。

リソースファイルをダブルクリックしてパターン一覧.Xが起動したら, すでにあるパターンをダブルクリックするか, メニューから「新規」や「編集」を選ぶと, お馴染みのパターンエディタが起動します。好きなようにパターンを描いたら登録ボタンをクリックしてPAT4形式で登録します。保存はパターン一覧の終了時に行われます。

こうして用意したパターンをピコピコエンジンで表示するには次の命令を使います。

●RESOURCE [リソースファイル名]

パターンをまとめてあるリソースファイル名をフルパスで指定します。普通は本体プログラムと同じディレクトリにリソースファイルを置くことにして,

"RESOURCE"+path$+"なんとか.LB"

のようにするのがいいでしょう。パターンの表示をするプログラムでは, 必ず一度この命令をピコピコエンジンに送らないといけません。

●PUT x, y, id

指定の座標に指定のID番号(パターン一覧でパターンの下に書いてある数字)のパターンを描画します。

●PUT@ x, y, id

PUTと同じですが, 表示画面にのみ描画します。

●ERASE x, y, id

指定の座標から指定のID番号のパターンが占める範囲を仮想画面から表示画面に描き出します。

おっといきなり表示画面とか仮想画面とか初耳な単語が出てきてしまいました。これはいったいなんなんでしょうか。

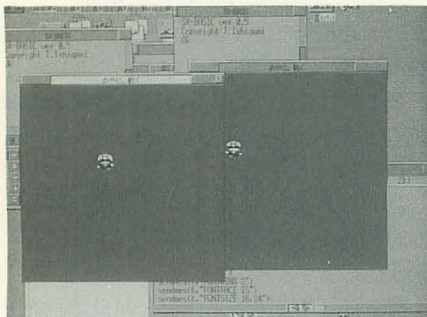
実は, SX-WINDOWのウィンドウ上に描かれた絵とか字とかは, 描いたら描きっぱなしで, システムはなにを描いたかなどは覚えてないのです。いままでほかのウィンドウの下に隠れていた部分が新たに見えるようになった場合, 当然その部分を描き直さねばなりません。描き直し方は各プログラムに任されているのですが, ピコピコエンジンではウィンドウと同じ大きさの仮想画面を用意して, 描画を表示画面(ウィンドウ上)と仮想画面の両方に行い, 描き直しが必要になったら仮想画面からコピーしてくるようになっています。

PUT@は表示画面にしか描画しないので, 背景を描いておいてその上にキャラクタを重ねて表示しても仮想画面には背景がそのまま残っています。そこで同じ座標でERASEを使うと, キャラクタを表示した範囲が仮想画面から描き戻され, 背景のグラフィックを壊さずにキャラクタを消去(移動)できるわけです。この方法だとウィンドウの描き直しが生じてもPUT@で描

リスト2 PRINT.SXB

```
 1: /*
 2: /*          文字表示
 3:
 4: int t
 5:
 6: t=fock("ピコピコエンジン.X "+str$(taskid))
 7:
 8: sendmes(t,"WINOPEN 文字の表示,268,40,1")
 9:
10: sendmes(t,"CLS 8")
11: sendmes(t,"FONTKIND 2")
12: sendmes(t,"FONTFACE IS")
13: sendmes(t,"FONTSIZE 16,24")
14: sendmes(t,"LOCATE 4,6")
15: sendmes(t,"COLOR 11,8")
16: sendmes(t,"PRINT LE SOLEIL DANS LE CO")
17: sendmes(t,"LOCATE 220,6")
18: sendmes(t,"PRINT EUR.")
19:
20: end
```

図1 キャラクターパターン

PAT4 #128
PAT4 #129
PAT4 #130
PAT4 #131
PAT4 #132
PAT4 #133
PAT4 #134

リスト3，4の結果

いたパターンは描き直されないのですが，そういうパターンは，きっと消しては描きを繰り返すようなパターンでしょうから，問題なしということにしときます。

キャラクタをただ動かしてみたのがChrMove.SXBです。ERASEで消して，位置をずらしてPUT@で描いて，を繰り返しているだけです。あとはキー入力ができれば，ゲームのようなものが簡単にできてしまいそうですね。

キー入力は?

キーボードやジョイスティックの状態を読み取るにはどうすればいいのでしょうか。inkey$はSX-BASICのウィンドウがアクティブでないと役に立たないし，キーバッファもたまるのでちょっと駄目です。iocs()関数でB_BITSNSを呼び出すのがいいのかもしれませんが，ピコピコエンジンを作ったときはそんなこと気がつかなかったので，タスク間通信でピコピコエンジンからキーの状態を送ってもらうという方法を使いました。

●JOYSTART *n*[, *f*]

この命令を送ると，以後n/100秒おきにピコピコエンジンから以下のようなメッセージがくるようになります。

"JOY 8246RABL, xxxx, yyyy"

8246RABL: ジョイスティック，テンキー，マウスボタンの状態を示しています。押されていないときは0になります。

例） スティック右上&Aボタンのとき

→80060A00

xxxx: マウスのX座標(10進数)です。

yyyy: マウスのY座標(10進数)です。

fに0以外の値を指定すると，ピコピコエンジンのウィンドウがアクティブでないときにもメッセージを送るようになります。

メッセージはgetmes()関数で受け取ります。注意することは，getmes()関数を実行するときは必ず割り込み許可状態(ei()が実行された状態)でなければいけないということです。そうでないとハングアップしてしまいます(ほかのタスクの実行を止めているのに，ほかのタスクからのメッセージを延々と待ち続けることになるので)。

getmes()関数の返り値が空文字列のときはまだメッセージがきていないので，メッセージがくるまでgetmes()関数を繰り返します。"QUIT"というメッセージがきたら，これはピコピコエンジンのウィンドウが閉じられたということなので，SX-BASICのプログラムも終了させます。"JOY～"というメッセージがきたら，メッセージ中のジョイスティック(テンキー)やマウスの状態を必要に応じて参照します。普通でないジョイスティック(チェルノブアダプタ&メガドライブのパッドなど)をつなげていると困るということに多少注意しましょう。

サンプルはOyajiMove.SXBです。テンキーでキャラクタが動きます。メッセージを受け取るところと，ジョイスティック状態を参照するところなどは参考になるかもしれません。こいつをちょっと拡張すればすぐゲームになりそうですね。

Oyaji2Move.SXBのほうはキー入力をIOCSコールでやってみたものです(テンキー操作のみになってます)。うーん，なんか格段に速くなりますね。こりゃ，こっちを使ったほうが確実によいような気がしてきたなあ。どうしよう。

リスト3 ChrMove.SXB

```
 1: /*
 2: /*        おやじが勝手に動く
 3: /*
 4: int t
 5:
 6: int x=0,y=0,vx=8,vy=8,c=128
 7:
 8: t=fock("ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ.X "+str$(taskid))
 9:
10: sendmes(t,"WINOPEN おやじ、動く,384,352,0")
11: sendmes(t,"RESOURCE "+path$+"OyajiMove.lb")
12:
13: sendmes(t,"APAGE 7")
14: sendmes(t,"CLS 14")
15:
16: while 1
17:         di()
18:         if x+vx<0 or x+vx>352 then vx=-vx
19:         if y+vy<0 or y+vy>320 then vy=-vy
20:         sendmes(t,"ERASE "+str$(x)+","+str$(y)+",128")
21:         x=x+vx:y=y+vy:c=(c+1) and 129
22:         sendmes(t,"PUT@ "+str$(x)+","+str$(y)+","+str$(c))
23:         ei()
24: endwhile
25: end
```

リスト4 OyajiMove.SXB

```
 1: /*
 2: /*        おやじがテンキーで動く
 3: /*
 4: int t,mx,my
 5: str mes[255],joystate
 6:
 7: int x,y,vx,vy,c=128
 8:
 9: t=fock("ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ.X "+str$(taskid))
10:
11: sendmes(t,"WINOPEN おやじ、動く,384,352,0")
12: sendmes(t,"RESOURCE "+path$+"OyajiMove.lb")
13:
14: sendmes(t,"APAGE 7")
15: sendmes(t,"CLS 14")
16:
17: sendmes(t,"STARTJOY 5")
18:
19: while 1
20:         ei()
21:         mes=getmes()
22:         if mes="" then continue
23:         di()
24:         if mes="QUIT" then break
25:         if left$(mes,3)="JOY" then {
26:                 joystate=mid$(mes,5,8)
27:                 mx=val(mid$(mes,14,4))
28:                 my=val(right$(mes,4))} else continue
29:
30:         if strchr(joystate,'4')>0 and x>0   then vx=-8
31:         if strchr(joystate,'6')>0 and x<352 then vx=8
32:         if strchr(joystate,'8')=0 and y>0   then vy=-8
33:         if strchr(joystate,'2')>0 and y<320 then vy=8
34:         sendmes(t,"ERASE "+str$(x)+","+str$(y)+",128")
35:         x=x+vx:y=y+vy:c=(c+1) and 129
36:         sendmes(t,"PUT@ "+str$(x)+","+str$(y)+","+str$(c))
37:         vx=0:vy=0
38:
39: endwhile
40: ei()
41: print "終了しました"
42: end
```

その他の命令

キー入力もできるようになって，あとはもうゲームを作ってしまうだけになったので，このへんで残りの命令を解説してしまうことにします。

●LINE x, y

現在のペン位置からx，yまで直線を引きます。ペン位置はx，yに移動します。

●BOX x1, y1, x2, y2[, rh, rv]

指定の長方形の枠を描きます。rh，rvを指定すると角の丸い長方形になります。

●BOXFILL x1, y1, x2, y2[, rh, rv]

指定の長方形を塗りつぶします。rh，rvを指定すると角の丸い長方形になります。

●CIRCLE x1, y1, x2, y2[, sr, er]

指定の長方形に内接する円を描きます。開始角度sr，終了角度erを指定すると円弧になります。

●CIRCLEFILL x1, y1, x2, y2[, sr, er]

CIRCLEと同じですが，中を塗りつぶします。

●SCROLL x1, y1, x2, y2, dx, dy

指定の矩形範囲内を，横dx，縦dyだけずらします。

●PAINT x, y

指定座標の周囲の同色範囲を塗りつぶします。

●EXPAT id

エクステンドパターンを設定します。idにはリソースファイルの中の16×16のPAT4データのID番号を指定します。これを実行すると，以後描画色の代わりに指定のパターンを使用するようになります。元に戻すときはidに0を指定します。

●PENSIZE h, v

ペンサイズを横hドット，縦vドットに設定します。普通は1，1になっていますが，太い線を描きたいときは大きくしましょう。

●PENMODE mode

ペンモード(描画時の論理演算)を設定します。modeに指定できるのは，

PSET	PSET
AND	NAND
OR	NOR
XOR	NXOR

の8種類(の文字列)です。

●FONTMODE mode

文字描画時の論理演算を設定します。指定はPENMODEと同じです。

●DRAWPAGE p

描画画面を制御します。pの指定は，

0 (意味なし)
1 仮想画面にのみ描画します。
2 表示画面にのみ描画します。
3 仮想画面，表示画面両方に描画します。

●VCOPY x1, y1, x2, y2

仮想画面の指定範囲を表示画面にコピーします。

●QUIT

ピコピコエンジンを終了させます。

おしまい

今回はピコピコエンジンのごく基本的な使い方をやってみました。

ついでにピコピコエンジンのデバッグ&機能拡張バージョンを掲載しておきますので，できるだけこちらを使用するようにしてください。リスト6をMAC.Xなどで打ち込んで(セーブサイズ2571バイト)，LHAで展開すれば新しいピコピコエンジンが使用できるようになります。今回使用したグラフィックパターンを圧縮したものをリスト7に示します。適当に同じようなパターンを作成しても構いませんが，自分で描くほうが面倒だという方はこちらを入力してください。手順は先ほどとほぼ同じです(セーブサイズ1567バイト)。それにしてもプログラムが長くなってくるとsendmes()が大量に出現してうっとうしくなってしまいますね。SX-BASICに特殊ビットマップが早くサポートされないかなあ。

リスト5 Oyaji2Move.SXB

```
 1: /*
 2: /*        おやじがテンキーで動く
 3: /*
 4: int t,mx,my
 5: str mes[255],joystate
 6:
 7: int x,y,vx,vy,c=128
 8: int i,j
 9:
10: t=fock("ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ.X "+str$(taskid))
11:
12: sendmes(t,"WINOPEN おやじ、動く,384,352,0")
13: sendmes(t,"RESOURCE "+path$+"OyajiMove.lb")
14:
15: sendmes(t,"APAGE 7")
16: sendmes(t,"CLS 14")
17:
18: while 1
19:         ei()
20:         di()
21:         i=inp(8):j=inp(9)
22:         if (i and 128)>0 and x>0   then vx=-8
23:         if (j and   2)>0 and x<352 then vx=8
24:         if (i and  16)>0 and y>0   then vy=-8
25:         if (j and  16)>0 and y<320 then vy=8
26:         sendmes(t,"ERASE "+str$(x)+","+str$(y)+",128")
27:         x=x+vx:y=y+vy:c=(c+1) and 129
28:         sendmes(t,"PUT@ "+str$(x)+","+str$(y)+","+str$(c))
29:         vx=0:vy=0
30: endwhile
31: ei()
32: print "終了しました"
33: end
34:
35: func int inp(x)
36:         set_reg(1,x)
37:         iocs(4)
38:         return(ref_reg(0))
39: endfunc
```

リスト6

```
000000  26 FA 2D 6C 68 35 2D E2 : 65
000008  09 00 00 E4 0E 00 00 01 : FC
000010  9A 32 1E 20 01 0D CB DF : C2
000018  BA CB DF BA B4 DD BC DE : 49
000020  DD 2E 78 10 D3 48 00 00 : AE
000028  08 D8 73 F7 BE D5 B4 A4 : 35
000030  B7 F7 BD 6B B5 D5 C5 7D : A2
000038  AC D3 75 44 9B AB 61 0E : ED
000040  3B 1B 36 9A 58 73 03 06 : FA
000048  BE F5 B7 6B 6B 8F 5A 5D : 86
000050  76 2B 82 66 F9 83 7C F0 : 71
000058  03 2C A6 A0 2F 13 14 F5 : C0
000060  23 0D C0 34 E5 E8 F1 3A : 1C
000068  31 9B D8 21 95 50 99 15 : 58
000070  55 79 E5 11 5D 1E 38 64 : DB
000078  38 8F 74 7B A3 38 D1 7B : DD
-----------------------------------------
CKSUM:  1E DE 4D CC 71 E2 0E 45 59A8

000080  80 58 19 39 98 C6 D7 5D : BC
000088  6F EF BE F5 D5 3A 40 27 : 87
000090  4E FE 1E FF 00 76 E1 9B : 5B
000098  8D B8 E5 7D 74 19 DF 35 : 48
0000A0  00 65 C4 0F FF F4 14 20 : 5F
0000A8  43 15 7E 18 95 A3 00 5F : 85
0000B0  95 DA CF 7D E4 32 AE A3 : 22
0000B8  55 44 2C 1F C0 97 B2 03 : F0
0000C0  79 69 07 73 CE 11 F2 9F : CC
0000C8  6D 02 B6 D6 D2 CF D3 E1 : 50
0000D0  EB 75 F3 6A DD 73 E8 26 : 1B
0000D8  06 F1 DA D9 02 F4 EA DB : 65
0000E0  82 BA 58 B3 70 70 F4 8E : A9
0000E8  80 1C 59 05 69 AE 83 BA : 4E
0000F0  E6 BA 71 A8 06 08 FE E0 : A5
0000F8  5F 1B 14 1B 19 08 1A 99 : 7D
-----------------------------------------
CKSUM:  15 11 D7 74 90 64 71 BB F2A8

000100  C5 06 A6 E8 C4 74 04 6D : 02
000108  06 3B 6A 6D B1 B6 C4 1A : 5D
000110  32 8E B2 0C 99 76 41 D7 : A5
000118  4E B7 43 64 12 13 64 DA : 0F
000120  C8 33 ED 4C E0 F1 83 67 : EF
000128  88 A6 C0 D4 65 69 0F 0D : AC
000130  9C 90 91 A8 46 0D 68 49 : 69
000138  30 83 F8 C2 4D 54 85 E8 : 7B
000140  C1 FB 1A 6C 17 9A 80 66 : D9
000148  F4 56 1F 5E 4B 02 62 FD : 73
000150  6B 1B 99 4B 16 84 9E E0 : 82
000158  98 BD 05 8D F8 3E A5 87 : 49
000160  CF 92 C4 98 BF E6 91 B4 : A7
```

```
000168  94 FE 0C 1F 42 04 B4 C8 : 7F
000170  4E EB 07 D1 A4 4F DC 8D : 6D
000178  DD 1C 67 78 F4 71 9B 4F : 27
-------------------------------------
CKSUM:  AD 32 50 F1 01 76 CD FF 7C61

000180  81 07 C1 6B 26 A4 F9 D5 : 4C
000188  7E D9 CD 3E AB CD 81 B2 : 0D
000190  1C 33 E4 EB 3F 85 1C 86 : 84
000198  73 C2 C7 07 0D C8 3E 74 : 8A
0001A0  93 42 16 E0 D4 8E 42 B0 : 1F
0001A8  8A 72 3F 12 40 C6 ED 1B : 5B
0001B0  D4 65 5F B8 36 58 05 1A : FD
0001B8  30 1D 6C AC 25 CF 5D D2 : 88
0001C0  E6 D4 B3 E4 A0 E9 F3 A9 : 76
0001C8  73 C3 61 D8 2D 89 0B 7C : AC
0001D0  E8 81 8B BB 46 53 C7 58 : 67
0001D8  9B 95 DD 2F 11 7C 80 8D : D6
0001E0  CB 16 CD 94 6E 69 75 D5 : 63
0001E8  16 CE BD 6E A0 57 47 B6 : 03
0001F0  80 3A 77 1D B0 E1 B6 61 : F6
0001F8  2C 0D 8E D1 BD 52 B4 22 : 7D
-------------------------------------
CKSUM:  18 E3 64 87 2B 6D D0 50 566D

000200  CD 6F 64 15 EA E6 C5 1D : 67
000208  A9 05 F1 EB B0 7D 24 D9 : B4
000210  93 6F DB D5 70 F0 A1 13 : C6
000218  0E A5 30 0E AD 3D D1 8B : 37
000220  AF 9A 8C A2 B5 61 62 71 : 60
000228  85 A7 02 9B 36 A0 8D D4 : 00
000230  F4 E2 7B 04 DC 91 A6 DC : 44
000238  30 BC 58 53 6C C1 3C F8 : F8
000240  E8 98 0C BE 09 05 53 E6 : 91
000248  A6 43 97 48 BD 9A 37 BF : 15
000250  F1 AE CC AD 71 A2 9C 2C : F3
000258  F2 40 1E DE 13 28 B9 D9 : FB
000260  2C 09 93 AD EA 06 4A 32 : E1
000268  8A EB AE 2B EC 00 D4 EA : F8
000270  C4 9A E7 CE 86 2D AC 4D : BF
000278  04 65 0D 2C B6 C4 DC 87 : 7F
-------------------------------------
CKSUM:  5E 23 83 DA 46 43 B1 47 A74A

000280  7A 9E 98 42 62 6E 04 26 : EC
000288  A7 61 89 B9 30 D1 92 F8 : D5
000290  22 6D BC 72 A8 E2 6D 03 : B7
000298  62 9B 38 CC 87 05 AC 76 : AF
0002A0  03 19 AF 26 87 A3 2B 89 : CF
0002A8  C4 1C DD 12 75 A3 55 BF : FB
0002B0  C4 BA 3F A7 57 5D DF 71 : 68
0002B8  83 54 AE BA 2C D2 C1 35 : 33
0002C0  03 93 3C A5 B0 21 78 3D : FD
0002C8  FC 59 B2 3B 87 73 98 F2 : C6
0002D0  12 06 1E 6A 18 7C 38 65 : D1
0002D8  5F 6C 28 BC C4 30 D0 4C : BF
0002E0  66 5A B2 7F 6D B4 05 ED : 04
0002E8  EB 1B 9C AE EA 0F DC 15 : 3A
0002F0  CA FC C5 96 76 81 5C DA : 4E
0002F8  15 BC EA AD 3E 8F C0 6D : 62
-------------------------------------
CKSUM:  53 D5 BF 48 5E AE E4 AE A4DE

000300  DE 26 C8 FC 83 81 26 31 : 23
000308  F2 1F 1E 88 92 8C A6 F5 : 70
000310  47 34 4E A3 13 BB 8E CE : 96
000318  36 49 DD DA 6C EA C0 2A : 76
000320  93 AB 10 B5 4E A2 45 16 : 4E
000328  DE 10 B0 7C 24 AA 10 96 : 8E
000330  8C A8 78 07 56 52 C9 C8 : EC
000338  2A 2E 3A 32 92 E9 A1 2D : 0D
000340  0D D9 D5 82 7D 94 76 10 : D4
000348  F0 12 5E F0 F1 B4 3A C4 : F3
000350  EC 21 E0 15 0D B1 53 F4 : 07
000358  02 1B 14 EB 97 8D CA 78 : 82
000360  8C 89 8F E3 30 E5 0D 2E : D7
000368  F3 9D 46 F9 D0 1A 57 7D : 8D
000370  D1 E4 B5 92 09 60 E8 5F : AC
000378  20 D0 56 D9 15 6E C5 3A : A1
-------------------------------------
CKSUM:  CF 54 8A 24 1E 8C B7 43 1D8D

000380  83 B9 30 CC 8D 8F 5B BB : 6A
000388  B8 E7 AF 5C 8C A2 6F 2C : 73
000390  0E 30 D2 1E D8 F2 6E FF : 65
000398  2D 25 1C 98 9E C5 20 A3 : 2C
0003A0  B9 D3 18 9F 86 23 BE 34 : DE
0003A8  F0 5D CE 3E CF 50 74 8E : 7A
0003B0  1D E8 F0 03 57 CD 76 35 : C7
0003B8  76 83 E7 42 59 DC 38 CD : 5C
0003C0  C2 D6 D2 8D 16 E3 FE 8D : 7B
0003C8  1B 53 08 5B EF 1A 88 13 : 75
0003D0  4D D0 C3 57 2C 74 C7 3F : DD
0003D8  B2 D0 69 1F 02 F4 3C 50 : 8C
0003E0  B6 6E C1 5D 2A 1A D5 78 : D3
0003E8  11 66 9F D4 1C FE 13 8E : A5
0003F0  53 89 88 DE A5 68 5D C3 : 6F
0003F8  D2 EA BD E0 A9 55 FC 59 : AC
-------------------------------------
CKSUM:  7A A0 35 4D 5B 3E 02 9E 546E

000400  B2 6A 92 0E E0 78 13 34 : 5B
000408  17 1C 3C 34 9C 90 50 EA : 09
000410  8C 17 5C 23 E9 0C 36 C8 : 15
000418  47 8C 30 9F 75 1C D9 35 : 41
000420  31 98 8B 24 ED 33 47 2D : 0C
000428  24 D4 8C 93 94 5D C1 A8 : 71
000430  5A 80 46 4F CA 17 1F C4 : 33
000438  E1 4B 53 06 13 CA B5 B4 : CB
000440  00 57 97 0A 69 38 F5 A9 : 37
000448  EF 86 13 D1 0C 1F CC 6B : BB
000450  08 49 B0 18 BE C1 57 19 : 08
000458  D0 E9 77 8E F6 17 50 32 : 4D
000460  07 FF 58 BA 47 94 2F 5D : 7F
000468  8B 93 35 27 94 98 DA 62 : E2
000470  13 38 49 BB B5 31 BD A4 : 96
000478  23 E2 39 E2 7A 7A F2 36 : 3C
-------------------------------------
CKSUM:  BB 1B EA 0F 6B A7 6E 60 1A09

000480  6C DA C2 8B BD F9 E2 FF : 2A
000488  48 BA 94 28 CF D4 BB A3 : BF
000490  42 D6 D1 70 11 FA DD 80 : C1
000498  D7 EF 59 33 55 50 47 90 : CE
0004A0  9B 14 81 7C 25 54 10 EC : 21
0004A8  09 C1 CD 27 0B 0C E7 0E : CA
0004B0  6D 0C E7 0E 6E 47 65 C3 : 4B
0004B8  A0 FF CE 15 54 FC 19 79 : 64
0004C0  BE F5 36 5F 22 9B 2F D0 : 04
0004C8  EC B2 EB 7B 4C B8 C0 6E : 36
0004D0  09 18 62 97 B9 72 63 E4 : 8C
0004D8  04 77 CE 4C 6E 6A BE 32 : 5D
0004E0  40 BB 04 A1 F5 C9 5F E4 : A1
0004E8  A3 52 51 7E 85 F0 A0 F0 : C9
0004F0  A4 01 D9 FB 50 43 0D BB : D4
0004F8  1A 53 1F EA CE 98 D7 6C : 1F
-------------------------------------
CKSUM:  D6 D0 21 DD 11 7D 29 37 776D

000500  D5 FD 42 19 74 9A 7A 37 : EC
000508  26 08 5B 40 3D 36 96 A6 : 78
000510  2A 2B 6B AE 0A 18 CB 50 : AB
000518  0C 59 B1 26 DA 9A AD B8 : 15
000520  E0 B9 28 5F 44 9A 45 79 : BC
000528  FD 5D 08 AB CF FB 46 97 : B4
000530  CD A4 E1 91 4D C2 C5 59 : 10
000538  02 1A F8 AB 24 E2 AD 39 : AB
000540  CF F7 D6 9C 67 FB 22 39 : F5
000548  E2 3E 38 8E 85 2D 1B FA : AD
000550  04 FF 3C 90 4C FE 13 83 : AF
000558  25 A1 1D B1 34 F6 93 5E : AF
000560  4E 73 5C DE 42 B2 44 38 : 6B
000568  A4 A0 94 65 11 F3 D8 45 : 5E
000570  DB 15 17 A0 39 FB F5 76 : 46
000578  C2 1C 18 DF D9 6D 75 A3 : 33
-------------------------------------
CKSUM:  46 76 48 A0 EA E4 EE 31 AE05

000580  7E 2D 54 0A DB B4 ED B6 : 3B
000588  F6 B4 DB 77 6A ED 44 3B : D2
000590  1F FA DB DB D3 6D B8 57 : 1E
000598  6E 21 79 4D B6 D9 20 CC : D0
0005A0  E8 11 B7 99 8C 56 DD 7F : 87
0005A8  61 56 F8 AA B6 29 01 93 : CC
0005B0  95 BE 15 56 51 9A F6 5A : F9
0005B8  FB 8C E7 E9 9F 42 BB 81 : 74
0005C0  0F 4F B1 4E FF 28 B4 F7 : 2F
0005C8  E5 6C F9 3D B6 CC AA 6D : 20
0005D0  9C 15 47 58 4C F5 FB 6D : F9
0005D8  9F 1E 9B 66 FD 5F 1C 43 : 79
0005E0  E5 F6 1B 23 46 39 96 65 : 93
0005E8  6C D8 A1 87 D9 57 BF A6 : 01
0005F0  AF 6C A8 E9 B9 97 8B 51 : D8
0005F8  2E 61 B0 0F 1B DB 5D F5 : 96
-------------------------------------
CKSUM:  37 36 CE 16 F1 8C 4A 66 9C2E

000600  78 37 43 70 25 0F E1 70 : E7
000608  75 E5 BE 16 B4 F9 A3 16 : 94
000610  69 3F F6 B9 A3 59 93 3B : 21
000618  DB 1A 65 7E 3D 47 F4 0A : 5A
000620  19 CB 52 6C EC 22 41 F3 : E4
000628  BC E2 41 AA F9 43 8F F2 : 46
000630  94 3F 4A D4 83 3B 78 2A : 51
000638  7B E1 3F 90 50 E2 AC A6 : AF
000640  66 8C 71 78 02 7F 9C 50 : 48
000648  E2 13 14 79 5B 09 CF 07 : BC
000650  9D DB 41 E5 5D 08 EB C0 : AE
000658  F0 30 49 7E 52 4D 12 6E : 06
000660  8A 4A F1 26 09 49 60 24 : C1
000668  DC 94 93 84 9F 02 3F C6 : 2D
000670  24 8E A5 AB BF 49 E0 FF : E9
000678  BC A5 0E AF 2C FB DC 92 : B3
-------------------------------------
CKSUM:  30 FD BE 8F 10 96 C2 80 78F8

000680  C7 D5 B7 D5 4B 1E 9A 57 : 82
000688  BE F1 46 C7 76 A3 C4 BD : 56
000690  BC 49 A9 34 CE C9 1E 2C : C3
000698  FB EA 78 BA 02 F6 DB A2 : 8C
0006A0  4D 99 34 66 2C DD 24 2A : D7
0006A8  5C 69 E9 49 2A 8D AE E8 : 44
0006B0  96 EA DC 6A 97 8A 90 65 : DC
0006B8  F9 43 77 7C 53 BF AC 79 : 66
0006C0  E3 A7 7C 5E 22 AB 69 6E : 08
0006C8  5F E9 90 0E 0C 1E 75 46 : CB
0006D0  D3 F2 EF FB C3 C3 AE 8C : 6F
0006D8  A1 F0 93 C7 25 F5 8E 0D : A0
0006E0  E1 2F BC 8A 6B 78 03 99 : D5
0006E8  EE 95 55 90 C1 AF 12 DE : C8
0006F0  0F FF DA B4 7F 97 67 24 : 3D
0006F8  0E AC C4 B4 04 63 AF 92 : DA
-------------------------------------
CKSUM:  16 09 CB CF 96 D5 AA 4C 9D3F

000700  BE 1E 01 F3 97 6D F4 2F : F7
000708  EF 97 36 80 38 A7 26 C6 : 07
000710  03 48 F3 F5 9C BA 3F 78 : 40
000718  F5 9E B7 64 38 F1 B2 60 : E9
000720  56 47 E9 F3 90 2A ED 41 : 61
000728  36 6F 2C 0B B3 6E C6 BC : 7F
000730  EC 4A 17 4B 6C DE 57 49 : 82
000738  6E 24 F2 B1 9C 98 AD E0 : F6
000740  F1 7E E9 B3 E2 D2 5B 36 : 50
000748  E6 93 49 DB 9C 3E 23 7F : 19
000750  02 BF A0 68 0E AC B4 B7 : EE
000758  9F 1A 31 FB 4B 97 A2 FF : 68
000760  79 63 8B 8E 26 EB A9 35 : E4
000768  7C B1 E2 3E 17 67 91 9D : F9
000770  51 39 19 90 76 24 58 EE : 13
000778  47 FB 87 7D 40 BE 91 F8 : CD
-------------------------------------
CKSUM:  90 F1 0F 90 B8 54 B9 16 B8FE

000780  DB 7D 1A CF 96 8C F1 D5 : 29
000788  7B E5 6D B5 5E F2 E0 55 : 07
000790  E4 59 A2 CB DE F6 F0 78 : E6
000798  23 7A A4 E9 B5 E1 3A 35 : 2F
0007A0  28 6F E2 2D E0 6A C1 AC : 5D
0007A8  23 FA 41 5A 84 BB A3 F6 : 90
0007B0  4C A2 6F 4E 68 C3 92 01 : 69
0007B8  FC 8B E4 C2 43 5B 05 CB : 9B
0007C0  A6 65 72 A1 58 3B 6E 18 : 37
0007C8  3D EC 59 BA 17 4D 61 93 : 94
0007D0  5D C7 61 F1 F5 23 E2 1B : 8B
0007D8  46 7D 94 0A 3D 54 38 E5 : 0F
0007E0  AF 9C 7B 5B 20 D8 F1 1A : 24
0007E8  44 96 9C E1 F1 8B BD 6E : FE
0007F0  C4 9E 1A A4 EF EF 75 B8 : 2B
0007F8  4B BF 81 9F 4E 23 68 09 : 0C
-------------------------------------
CKSUM:  78 EF B5 A4 85 0C 6A 39 3C88

000800  07 5F 13 50 78 6C 64 05 : 16
000808  C7 3E BE 81 68 9D 84 7A : 47
000810  5E 1F E8 26 DE A9 0C 3C : 5A
000818  2F CE 3A 2A C4 67 55 E3 : C4
000820  D8 5B 01 F1 89 B4 0D 4B : BA
000828  5D 5A 1C 22 40 E3 46 72 : D0
000830  9F 79 06 7C 43 27 99 85 : 22
000838  93 AF 9B 18 32 78 42 7F : 60
000840  5A 36 3B EE 8E 34 B1 84 : B0
000848  F2 46 85 F2 6F AA 7E CE : 14
000850  13 30 34 C0 EE AA 0C 07 : E2
000858  15 66 06 9F C4 36 F4 C3 : D1
000860  BE 11 DE 9E 92 07 0C 4F : 3F
000868  93 9C BE 8D ED C7 3C CA : 34
000870  94 65 65 7A 81 2F 2D 90 : 45
000878  6F 32 EF 03 03 2F CF 09 : 9D
-------------------------------------
CKSUM:  8A BD 9B AF 72 39 EA 2D EC8A

000880  B9 7E F8 79 B9 7F 78 79 : D1
000888  D3 DC 19 02 77 34 3D 1C : CE
000890  CC 60 C2 9D 60 1E C4 ED : BA
000898  B0 7A B9 96 21 EE E6 79 : E7
0008A0  21 F0 D0 DE 87 57 37 14 : E8
0008A8  32 67 EE 03 99 3F CD 0F : 3E
0008B0  E7 3F 94 1F DE 7F A8 1D : FB
0008B8  1A 15 A1 42 85 99 95 13 : D8
0008C0  DA 18 7C FF 0C CA BC DA : D9
0008C8  E3 19 E6 ED 8C AF CD E0 : B7
0008D0  99 F9 33 78 A6 38 EB EF : F5
0008D8  4C AE CD F4 8C FC D9 56 : 72
0008E0  06 42 C9 86 65 A1 3D 41 : 1B
0008E8  61 5D 44 27 F7 40 F6 19 : 6F
0008F0  F7 2F 42 E2 0B B3 E2 3D : 27
0008F8  08 50 DF A3 10 1D C1 B8 : 80
-------------------------------------
CKSUM:  64 D5 0F 7A 75 CB C3 9C 8EEC

000900  83 0C 2E 5D C3 83 71 70 : 41
000908  08 41 8B 6C FD 61 85 C3 : E6
000910  F4 5E 83 B7 F0 DD DC 67 : 9C
000918  08 B9 56 D0 51 89 72 FE : 31
000920  F1 E9 48 F0 1F A2 F0 A4 : 67
000928  5B 1E ED 6C 1D 5C 17 8F : F1
000930  41 EC 51 A5 D0 84 F5 12 : 7E
000938  9A 23 2A 84 23 C8 EF 3E : 83
000940  11 F6 E2 D0 B8 B9 0D 6B : A2
000948  F4 60 8E 58 43 7B 73 06 : 71
000950  EA 18 AF DC C4 3E 1C 41 : EC
000958  F4 93 07 A8 BC 20 B5 6F : 36
```

▶映画「パルプフィクション」が近くでやってない。3月までおあずけのようだ。でも，「レザボアドッグス」のほうがいいという声もちらほら。　道越　秀吾(15)山口県

```
000960  51 BA 0D 65 D3 F8 80 F2 : BA
000968  19 FA D2 AB DD BB 83 0B : B6
000970  C1 0B B8 67 EA EE 5E C4 : E5
000978  0E FE CD CE 9A 19 E8 5C : 9E
-------------------------------------
CKSUM:  CA 38 CC C6 DF E0 C9 59 60B2

000980  04 51 90 87 F1 E3 06 C2 : 08
000988  26 44 CB 79 98 1B BD 72 : 90
000990  9E 97 EB E3 70 7D 95 3D : C2
000998  85 3E 91 FF FB F7 68 4D : FA
0009A0  7F FB 3D C2 9B ED 02 92 : 95
0009A8  54 DB CC BB 99 6D 33 7D : 6C
0009B0  37 EB 53 E4 DF 23 E6 71 : B2
0009B8  C7 23 E3 53 8C A7 BE A7 : B8

0009C0  AC A6 16 79 88 29 EA A9 : 25
0009C8  CC 99 B6 99 C0 99 06 66 : 79
0009D0  04 C9 6A 7C DE 5C 2F 33 : 4F
0009D8  15 4F 91 4F 9B CA E4 F0 : 7D
0009E0  36 A2 3F 4C 58 BE E2 9F : FA
0009E8  EB 3D A3 B9 8B 64 81 F7 : EB
0009F0  2F DD A9 E8 A9 D1 E0 E1 : D8
0009F8  EF FF C0 AF 7D AA 7D 93 : 94
-------------------------------------
CKSUM:  EE 60 28 0F 5D 1B 5C 21 EA15

000A00  2F 66 45 98 88 A3 FB 5F : F7
000A08  94 00 00 00 00 00 00 00 : 94
000A10  00 00 00 00 00 00 00 00 : 00
000A18  00 00 00 00 00 00 00 00 : 00
000A20  00 00 00 00 00 00 00 00 : 00
000A28  00 00 00 00 00 00 00 00 : 00
000A30  00 00 00 00 00 00 00 00 : 00
000A38  00 00 00 00 00 00 00 00 : 00
000A40  00 00 00 00 00 00 00 00 : 00
000A48  00 00 00 00 00 00 00 00 : 00
000A50  00 00 00 00 00 00 00 00 : 00
000A58  00 00 00 00 00 00 00 00 : 00
000A60  00 00 00 00 00 00 00 00 : 00
000A68  00 00 00 00 00 00 00 00 : 00
000A70  00 00 00 00 00 00 00 00 : 00
000A78  00 00 00 00 00 00 00 00 : 00
-------------------------------------
CKSUM:  C3 66 45 98 88 A3 FB 5F CF68
```

リスト7

```
000000  25 CC 2D 6C 68 35 2D F7 : 4B
000008  05 00 00 58 0D 00 00 9D : 07
000010  B5 32 1E 20 01 0C 4F 79 : FA
000018  61 6A 69 4D 6F 76 65 2E : F9
000020  6C 62 AD B5 48 00 00 05 : 7D
000028  55 73 83 ED 5B 4B A9 7F : 06
000030  BD BD 8D EB B3 B5 FA B0 : 04
000038  08 52 14 BE AF F1 88 47 : 9B
000040  80 2F 03 21 48 58 7A 89 : 76
000048  08 72 6C C7 58 9C 65 D0 : D6
000050  E2 38 E9 16 2D 61 4D EF : E3
000058  3B 09 9E 06 18 E4 78 34 : 90
000060  76 87 09 A6 08 A4 5E 12 : C8
000068  2C 66 FA CD A3 88 C6 01 : 4B
000070  FF 81 89 C0 26 F5 73 01 : 58
000078  8F 01 BD B0 2B EC 6F AF : 32
-------------------------------------
CKSUM:  9B 9D C4 63 CB EE B6 F5 0D81

000080  BF 7E FB D7 D6 ED 81 30 : 83
000088  8B C2 B2 04 8F 03 0E 06 : A9
000090  47 A1 E4 39 17 66 56 D8 : B0
000098  E3 73 00 04 52 2A 05 5A : 35
0000A0  B2 23 1A E3 73 73 E7 07 : A6
0000A8  2F 93 BD 2E B6 88 AD 57 : EF
0000B0  BF 95 D2 F2 CA 9F F0 85 : F6
0000B8  84 20 E5 E1 A6 EF 84 1E : A1
0000C0  C5 6E FF 83 64 5C F0 C2 : 27
0000C8  F2 BF BF 8E 01 63 CC 0B : 39
0000D0  DA FD E9 4F 73 43 89 5B : A9
0000D8  BF 10 91 30 8E 1E CD 6E : 77
0000E0  FE 4F 30 02 D3 9C 1C 57 : 61
0000E8  F0 3F B4 5D F3 C0 2D 7A : 9A
0000F0  2F F6 C8 2F 8C E0 24 A8 : 54
0000F8  22 D0 30 80 50 EA B5 65 : F6
-------------------------------------
CKSUM:  27 4D 33 9A 6F 4F 26 DD 32D1

000100  1A 54 E8 51 A5 44 89 16 : 2F
000108  4C 28 91 A5 42 91 1A 55 : EC
000110  8C 82 5A 48 75 7B 1F 29 : E8
000118  E6 78 9B 11 1F B1 27 04 : 05
000120  47 3B 7C F3 3D A6 4E AB : CD
000128  D7 60 17 6A 73 B5 2D 2B : 38
000130  53 15 E3 54 C5 A4 C3 B8 : 83
000138  34 99 11 A5 7C F0 BD 7C : 28
000140  22 7E 29 27 E8 08 80 33 : 93
000148  96 D0 33 F4 1C 18 F9 8E : 48
000150  0B 89 EB AF DC 37 79 7F : 39
000158  12 47 71 7F 42 3B EB FD : AE
000160  80 8E 3A 57 F7 E2 39 99 : 4A
000168  28 63 3C 5A 50 52 BF DB : 5D
000170  63 EC 30 15 CD B7 78 D9 : 69
000178  B6 F4 98 EC ED 29 32 F3 : 69
-------------------------------------
CKSUM:  13 AE EB A0 8F 96 63 1F 9EF2

000180  24 BF 67 F2 5F E6 12 FF : 92
000188  EE 98 0C 7D F3 03 56 C9 : 24
000190  C2 B5 27 03 A6 B5 77 20 : 93
000198  1C 8B C6 B5 1B 9D DC CB : 81
0001A0  4E 45 E6 11 73 B8 2E 78 : 5B
0001A8  7A B9 B3 6E 2D 2C 30 B0 : 8D
0001B0  F5 4A 05 C4 22 6C 11 00 : A7
0001B8  22 3F 6F 4B E1 F3 70 C1 : 20
0001C0  49 B3 4B 9F B8 01 6D CC : D8
0001C8  49 D9 95 87 54 9F 4D C1 : 3F
0001D0  70 27 80 17 15 D5 10 D2 : FA
0001D8  A4 27 47 68 D2 46 9B 7F : AC
0001E0  2E 6B 08 B6 61 17 17 11 : F7
0001E8  75 25 81 14 77 CA 5B FE : C9
0001F0  48 8E 78 95 13 6B 7B 6C : 48
0001F8  23 80 A9 73 2B D8 B1 57 : CA
-------------------------------------
CKSUM:  83 96 BE 2C BF 5D 9D 4C BB01

000200  C6 9A 78 89 DE AF 9F C9 : 56
000208  4E C9 37 76 06 AB 7F FE : F2
000210  60 1F 8F 1A 55 9E 14 69 : 98
000218  5A 8C 27 0C A6 52 18 C8 : F1
000220  26 B6 F2 56 69 C0 E6 79 : AC
000228  97 35 8D 29 F6 DA 93 13 : F8
000230  28 E8 9C 98 07 66 A6 29 : 80
000238  EF 6A 62 B9 7B 9C BA 2F : 74
000240  4A CE C8 B3 4C B0 DD 32 : 9E
000248  CB F7 40 23 1D 4F C5 E0 : 36
000250  57 94 47 D1 40 2A 2E 87 : 22
000258  86 01 EB 2F DB 70 CB 65 : 1C
000260  3C CE 45 6C E6 92 BB CC : BA
000268  BA B9 D5 76 BA FF 1F 60 : F6
000270  D3 39 B6 E7 BD CD B7 72 : 5C
000278  F7 B3 B4 45 EB CF E2 DE : 1D
-------------------------------------
CKSUM:  54 18 A0 D9 8C AC 31 56 AAB1

000280  E2 FE FD E0 B0 F4 88 FB : E4
000288  E4 BF 7B 0C 3C 50 0F A8 : 6D
000290  B8 27 FE D6 C8 3B 74 D3 : FD
000298  F6 FB 0C 4C 4C 2D 28 07 : F1
0002A0  F5 08 EC 8A 0A 6E 6A D9 : 2E
0002A8  2D 08 EA A0 E6 00 53 0A : 02
0002B0  A1 2D 94 4C 0C 27 87 82 : EA
0002B8  01 D2 FA F1 44 6A 0E 8D : 07
0002C0  57 CF F0 41 3A 6D 69 B9 : 20
0002C8  A6 F6 5E 73 95 8D C2 28 : 79
0002D0  1A B6 71 2C FE 25 EB 65 : E0
0002D8  FC FE 24 DE DF 30 37 FA : 3C
0002E0  D0 0E 4B B0 EA E0 82 9D : C2
0002E8  25 81 56 C9 C1 B4 A3 4B : 28
0002F0  02 8D 1A D5 43 DA FD 72 : 0A
0002F8  1B B6 9F 31 BB 37 1A 9F : 4C
-------------------------------------
CKSUM:  5D 39 23 B2 95 9F 0E A8 DD9D

000300  09 B0 9E 52 0F 26 15 84 : 77
000308  39 31 43 59 26 35 8A F3 : DE
000310  E2 C2 E3 73 62 87 FE 95 : 76
000318  18 01 F3 2E 24 E9 52 A2 : 3B
000320  9E 34 93 1E 34 E2 D1 D0 : 3A
000328  8C 39 84 77 44 77 80 73 : 6E
000330  6F 14 3E C6 DC CD FA BB : E5
000338  EC 1C EF C3 7B D3 5F A9 : 10
000340  26 82 FE DB EB FC E6 41 : 8F
000348  4C 81 05 70 5F 60 6F 93 : 03
000350  18 4F 6B 95 4D BB 6E 0B : E8
000358  4D D9 E6 12 FE 99 2F D4 : B8
000360  25 FA A4 BF 54 97 EB 2F : 87
000368  3E 39 B8 C4 C1 E9 BB AE : 06
000370  26 16 93 B7 CB C1 77 6B : F4
000378  D6 61 0C 38 C2 0B 3F A2 : 29
-------------------------------------
CKSUM:  F7 16 4A CE C1 C0 E7 F2 0510

000380  B3 2F 80 8B 3A 2D 0E 82 : E4
000388  20 5F F8 69 AE FF 92 17 : 36
000390  EB 6B 88 59 C6 29 4B 11 : 82
000398  8E 1E 2A 4E 7F 29 A7 27 : 9A
0003A0  4E 4E 5F 47 3F 93 CF 35 : 18
0003A8  58 0F 7E 04 F4 8A FA 05 : 66
0003B0  DD 22 C6 8E 87 7F D6 7F : AE
0003B8  4C 03 D7 90 F2 C1 EC 16 : 6B
0003C0  86 F9 82 75 1E 59 C5 5C : 0E
0003C8  55 D5 9B 07 4C 12 28 AE : 00
0003D0  0B 60 4B 93 7F F3 C9 85 : 09
0003D8  0E 1C FB 13 7C B3 81 6E : 56
0003E0  29 67 15 AE 9A 05 B2 A6 : 4A
0003E8  4C F2 31 B8 81 A9 FF FC : 4C
0003F0  E8 9B AC 7F 24 15 6A 0F : 60
0003F8  B0 B4 0A CD D9 82 DD 18 : 8B
-------------------------------------
CKSUM:  1C 8B 03 D8 56 31 4C 66 0F4D

000400  A0 B1 E8 D9 1A 73 C6 E3 : 48
000408  7A 79 7C 2F 75 9E B6 62 : C9
000410  18 54 EE 69 AF C5 93 A9 : 73
000418  4C 5C 56 FC 5D 5B 04 5D : 13
000420  86 2D D1 5C 6F B0 25 B0 : D4
000428  4B F3 1A 1C F5 89 7E A9 : 19
000430  2F D5 05 B7 F8 CD 2D 95 : 47
000438  8C 73 65 5F 1D 7E C7 41 : 66
000440  C6 51 1C 6B B0 1C 9D 22 : 29
000448  5A 8A 65 E9 4A 5E 5E 7B : B3
000450  DA 6B FF 1D 27 62 35 5A : 79
000458  EF 15 3F 8B DD 7C FE 4D : 72
000460  FC 9F 19 40 9B AB 55 9E : 2D
000468  B4 2F A1 B0 5E 9B D7 FE : 02
000470  EB F4 CF E9 80 75 6A FE : F4
000478  2F 55 2D AD 94 BF 7C CE : FB
-------------------------------------
CKSUM:  BD B4 72 7D 1F 87 EA 26 FAE7

000480  53 21 B3 DD 0B 87 A4 F1 : 2B
000488  1C 59 E7 38 D0 6C 72 68 : AA
000490  B8 3D 5E 1A 3A CE 31 D3 : 79
000498  F5 DB 63 AF 2C 17 EB C7 : D7
0004A0  05 FA 84 63 C3 08 D7 66 : EE
0004A8  08 A8 34 88 48 5E 7C 67 : F5
0004B0  65 CA 94 8A 11 A7 D1 72 : 48
0004B8  14 8B B2 54 3E 5D 89 22 : EB
0004C0  ED E8 92 10 C9 B1 9D 61 : EF
0004C8  B6 13 1A ED 27 49 BF FA : F9
0004D0  A4 14 D3 CF 4E 93 32 8B : F8
0004D8  F3 1A 5A C0 42 91 A3 35 : D2
0004E0  72 2A 9E EC D5 FE F9 FE : F0
0004E8  02 71 75 B2 70 C5 AB 5D : D7
0004F0  44 41 0D 29 DC 5D E5 EA : C3
0004F8  11 CE 2D E3 E9 3E 50 63 : C9
-------------------------------------
CKSUM:  A5 5C 7F DD 25 BE E9 17 5166

000500  FA C8 5D AD FB AC 1F 21 : B3
000508  0C B6 B2 0A 74 5E 29 D9 : 52
000510  22 BD 52 DE F9 0C B8 C2 : 8E
000518  66 CF 43 3F FF 8B F8 09 : 42
000520  AA 94 41 06 6E D6 D0 3E : D7
000528  0F C0 DE 91 4F 9F B8 BF : A3
000530  AD 00 EB 5E 04 C8 FE 78 : 38
000538  4E 97 64 1B DA 36 41 46 : FB
000540  8C 70 DC 5D CB D9 B0 CF : 58
000548  0D 1A E6 78 4E AF B5 17 : 4E
000550  35 DF 09 1D BF D4 DE B7 : 62
000558  3F E4 A7 AF ED 7C 74 B5 : 0B
000560  BC 0D 4E 46 B7 81 F4 F3 : 7C
000568  F5 FC 1F 31 ED C4 C3 3D : F2
000570  B8 0B 6E 9C B0 57 02 89 : 5F
000578  38 1A 0D EB E8 50 72 5C : 50
-------------------------------------
CKSUM:  F0 70 6C 83 03 D8 A1 E7 5A70

000580  B9 F2 64 C6 23 80 4E 36 : FC
000588  5A 4C A8 53 25 4B 59 A7 : 11
000590  58 05 4B F0 12 66 D6 1C : 02
000598  96 82 B6 D3 FF ED 36 50 : 13
0005A0  0E E4 B2 DB 06 4D 53 5F : 84
0005A8  5C A2 DA 85 6F CF 5E 4F : 48
0005B0  DB 3F AE 9C FE C1 6B 07 : 95
0005B8  8A 45 81 6C 9B 1D 32 34 : DA
0005C0  88 E4 11 E7 AF 03 8F 2F : D4
0005C8  D5 4A 18 EC 74 10 CD D2 : 46
0005D0  EE B2 A2 AE 96 DC A8 BB : C5
0005D8  05 BB 65 86 8C 50 3E 47 : 0C
0005E0  B2 BB 86 A8 2D A7 5F B7 : 85
0005E8  F3 FF 8D 7F A1 B0 5B B0 : 5A
0005F0  14 77 4B 1A 1B 1B A6 00 : CC
0005F8  F4 17 EF C9 18 44 9B 37 : F1
-------------------------------------
CKSUM:  CD B2 45 55 AD 0D 3E D3 9089

000600  13 11 26 E6 0E 73 7D 0A : 38
000608  BC F7 70 77 C1 E5 9B CA : A5
000610  83 9E 27 BD 57 9F C5 5B : 1B
000618  BF C0 7C 3A 53 80 00 00 : 08
000620  00 00 00 00 00 00 00 00 : 00
000628  00 00 00 00 00 00 00 00 : 00
000630  00 00 00 00 00 00 00 00 : 00
000638  00 00 00 00 00 00 00 00 : 00
000640  00 00 00 00 00 00 00 00 : 00
000648  00 00 00 00 00 00 00 00 : 00
000650  00 00 00 00 00 00 00 00 : 00
000658  00 00 00 00 00 00 00 00 : 00
000660  00 00 00 00 00 00 00 00 : 00
000668  00 00 00 00 00 00 00 00 : 00
000670  00 00 00 00 00 00 00 00 : 00
000678  00 00 00 00 00 00 00 00 : 00
-------------------------------------
CKSUM:  11 66 39 54 79 77 DD 2F 3E91
```

▶某誌のSEGA SATURNの記事中に「CPUの名前はX68000」とあった。あの小さな(?)SEGA SATURNにマンハッタンがつまっていたとは……。　西浦　宏和(20)愛知県

猫とコンピュータ

『考えること』を考える宿題

Takazawa Kyoko
高沢 恭子

どんなに忙しくても，東京と三重を行き来しているときは，できることに限りがあります。でも，いろんなことに思いをめぐらせるには格好の時間のようです。そうして思いをめぐらせた結果は……。

暮れの12月20日，夫よりひと足さきに三重から帰京した。いちばんたいせつに持ち帰ったのは，こんども，東京でやるべきことを記したメモだった。

書かれていることは，たとえば，○月○日までに提出をもとめられている書類の名称。数件，未済になっているお歳暮の送り先。○○銀行に出向く目的および持参するもの。親類のお子さんへのXマスプレゼントのこと。会う予定をしている人との連絡方法。

三重を出る前につくったこのメモを新幹線の車中で見なおした。

よく見ると，やらねばならないことと，やりたいと思うことがマゼコゼに書かれている。

丸善か伊東屋に行って，これとこれを買う。日本橋の刃物の専門店「木屋」に行って，よい包丁をみつける。新宿の画材屋でこれをさがしてみる。

お風呂場の壁を塗る。トオルのショルダーバッグの止めガネの修理を，◇◇ブランドの専門店に問い合わせてみる。食卓のイスをひとつ購入する。などなど。

これらはどれも，ぜひともやりたいと思って書きとめたことらしいが，東京に到着するころには，メモの内容に対する見解がだいぶ変わっていた。いくつかのことを除いては，どれも取るにたらないことばかりではないか。

予定や計画は，どこにいてそれを考えたかという，状況の影響があるらしい。

東京にいればあまり必要とみとめないことでも，遠方にいると距離のぶんだけ過大に感じて熱望してしまう。それが，すぐに手のくだせる現場にきてみると，平凡な価値に戻っているというぐあいに。

半年待ったマウス

意欲に満ちていたメモは，三重から東京までのあいだに色あせてしまったが，けっきょく取るにたらない計画のほとんどを私は実行した。

食卓のイスを1脚ふやしたくて，家具の専門店やスーパーを何軒かまわった。

いくつかの「気にいる条件」があった。スペースをとらないこと。しっかりできているが重くないこと。すわり心地がよいこと。マシンルームでも使えるデザインであること。好ましい感じがすること。

ありきたりの条件に思えたが，そうではなかった。食卓とセットになっているのがふつうだから，1脚ずつ売ってくれるものの中からしか選べない。

その中で「気にいる条件」をチェックしていくと，たぶん該当するものはゼロになる。なのに，ちゃんとイスを1脚手に入れたのだ。

ほかの予定もこなすとなれば，イスにかかわる制限時間がある。すべての条件を満たさなくても，なかの1脚を「最善」ときめたのだろう。判断は時間の制約に合わせて，都合よく調整できるものらしい。

そのとき思い浮かべたのが，三重のパソコンショップのY氏のことだった。

昨年の夏，DOS/Vパソコンの購入にあたりY氏のお世話になった。各部の仕様をきめるときはアドバイスを受け，部品のとりよせもおねがいした。

7月にマシンが到着したときに，Y氏は，キーボードとマウスはまだショップに入らないから待ってほしいといった。それまでの代替品に，Y氏のショップの備品であるものを置いて帰った。

9月ころキーボードが届いたが，マウスはいっこうに知らせがない。代わりがあるから困らないものの，気になって何回かたずねると，返事はいつも「もうちょっと待ってください」だった。

なぜマウスひとつでこんなに待つのか，フシギでならなかった。

12月の中旬に入って，やっとY氏から連絡があった。半年近くがすぎたことになる。「すこし前に品物が入って，テストをしてました」。夫と，クルマで40分ほどのY氏のショップまで，受け取りにいった。

「このマウスはいいですよ」とY氏。

いままで使ってきたいくつかのマウスにくらべると，ひとまわり小さくて，すべてが丸くなめらかだ。そのまま手のひらの丸みにふわりとおさまる。

聞けば，いろいろなマウスを検討した結果，この品がいいときめて，入荷を待ちつづけたのだという。Windowsではキーボードよりマウスが主役である。毎日使う道具として，よいものを供給したい。ほかの事情もあったのだろうが，そのために半年かけた。マシンについてのアフターケアをきちんとしたいと，彼がいっていたのをあらためて思いだした。

「直線を引くときに，とちゅうでドットがずれることがあるでしょう。このマウスはそういうことが少ないです」。よく吟味したつもりなので，とくに描画ソフトを使う私の感想を聞きたいとY氏はいった。

キーボードやマウスについて，こんなに慎重に接した人がほかにいたろうか。

そういえば彼は，CRTやスピーカーについても，折にふれて繊細な感想をのべていた。どれも目や耳や手など，人間の感覚に直接，微妙な影響をおよぼす機器である。パソコンを日用品とするなら，いちばん注意深く選ばれるべきものだった。

制限時間内でものを選ぶ私とは正反対に，Y氏はマシンにたずさわる人らしい姿勢を見せてくれた。

冬休みのレポート

トオルの冬期休暇の課題のなかで,「心理学フロンティア」の西川先生が出されたのは,「機械は考えるか」というテーマのレポートだった。

「おかあさんはどう思う？」とトオルに質問された。

「機械が考えるわけないでしょ」といったとたんに,「考えない」と判定する根拠や証明があやふやなことに気づいて,「ちょっと待って……」となった。

あれは電気とデータ入力のもとに動かされているのだ。「考える」ことはない。直感としてはそう思ったけれど,もし人間が「考える」ときと変わらないことをしているとしたら，レベルの差は別として，「考える」ことになりはしないか。

それなら，人間の「考える」ということはどんなことをしているのか。そもそも「考える」とは何か。自分なりに分析し列挙していくうちに,「考える」行為のもつそれぞれの要素が，たがいに関連しあって分類することさえむずかしくなってきた。そして，この作業自体，自分自身がつちかってきた「考える」力にもとづいたものであると気がついた。

しかも私が考える人間の「考える」は，機械にもそっくりのことができると考えざるを得ないような要素ばかりだ。このままでいけば,「機械は考える」と結論しなくてはならない。自分はそう考えたくないだけで，じつは機械は「考える」と考えるほうが正しいのか。

「その先生はどちらの考えが好きなの？」

「授業では，心はコンピュータというのがログセだけどね，どうかな」トオルが答えた。そして,「レポートはエッセイになってはダメというきびしい一点があるんだ。つまり証拠をしめして，結論しなくてはいけない」といった。

機械に聞いてみたい

国語辞典では人間の「考える」をどう説明しているか。新潮社，旺文社，小学館などから，すべて書き出してみた。

それぞれ5つから7つくらいの項目に分けて，独自の順位でかかげてある。

私はすべての点において，自発性をもたない機械は「考えない」と結論した。

トオルはテーマの検討の前に「考える」の定義づけをするとして，やはり，辞書の語釈を引用した。これは主観をさけるためだそうだ。

彼は多くの語意を5つに大別した。

「頭をはたらかせる」(判断)

「企画・創造する」(決心，立案，くふう)

「推測する」(想像)

「学習する」(反省，普遍性の追求)

「思いめぐらす」

これらの4番目までは，模倣であれ，同様の作業ができることを，彼は例をもってしめした。また，多くの場合，人間より有益な「考え」を展開しそうでもあると。

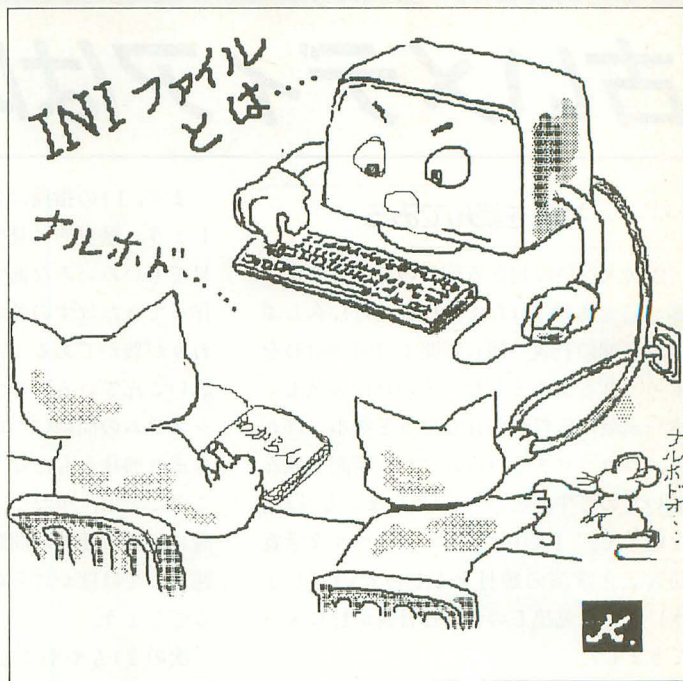
しかし機械の「考え」は人間の入力が前提であり，得た結果が人間に還ることで，従属的な存在である。機械は「考える」ことにみずから喜びをもとめないと。

そして5番目だけは異質で，不可能であろう。目的のない無意味な連想や，非論理的な発想をふくむからという。いつか，最高水準の思考回路をもつコンピュータが開発されたとしたら，この質問を「考えて」答えられるか。「機械よ，キミは考えているのか」と結んでいた。

レポートをつくり終えた翌日，トオルは西川先生の著書の1冊を買ってきた。

「序章だけでも読んでみたら」という。

冒頭になげかけられた質問は,「人間と機械はどうちがうか」だった。そして，どのような結論を得るかが問題ではなくて，それを導く「背景」になにがあったかを問題にしたいといわれていた。

それについて考えるために，なにを前提とするか。さらに，その前提がどんなものかを考える必要がでてくる。

illustration：Kyoko Takazawa

この「前提」を「常識」とよぶとき，常識が明確に自覚されていないにもかかわらず，それがなければ結論が得られないといい，この「常識」と機械とのある関連についてを指摘することで，冒頭の質問に対する西川先生自身の回答があった。

また，機械と人間との対比についても，1963年ころ，おもにソフトを介してコンピュータの研究にたずさわったときの体験をもとに語られていた。これも機械の特質を指摘していて痛明だ。

たとえば,「文章の誤りの部分を無意識に修正していくといったような人間の特徴はとても機械にはあり得ないことだ」。

私自身も，イスを選んだときのような，不合理な解決やあいまいな判断が，かえって便利で円満な結果を得られたと思うこともすくなくない。

「人間的なやりかただ」「機械的にこなす」などという。それぞれ，ばあいによって適否はわからない。時とばあいで判断に変動のあるのが，人間の特徴でもある。

私のメモが，新幹線のなかで西に向かううちに，また輝き出すかもしれないと思うのは楽しい。

参考文献：西川泰夫著：『認識』のかたち
誠信書房刊

第90回　知能機械概論――お茶目な計算機たち――

古いメディアは叫ぶ

社説を読んでみる

阪神大震災の起きる前日のことですが，振り替え休日だった1月16日（月），久しぶりに新聞の社説（朝日新聞1/16付朝刊）をじっくりと読みました。その日は入試センター試験の監督が前日でやっと無事に終わり，ゴロゴロとくつろぎながら新聞を読んでいたのです。そこへ，『あいまいな「私」を越えて』（大江健三郎のノーベル文学賞受賞記念講演の題目をもじったのでしょう）という見出しのついた社説が目に入ってきました。

最近，安心して新聞を読めなくなっており（理由はあとで書きます），しかも，その内容が計算機による新しいメディアを批判する内容のようでしたので，ことさらきちんと読む気になったのでした。

入試センター試験の問題でこそ過去に取り上げられているかどうかは知りませんが，朝日新聞の天声人語とか社説とかいうのは，入試問題によく使われてきたようです。ひとつの模範というかそれなりの完成度をもつ文章が載っているものとして，一般に認められていることを物語っているのでしょう。

しかし，この日の社説についてはどうにもしようがありません。主張している内容に賛成か反対かという以前に，ひとつの文章を構成するに必要な一定の水準に達した論理がないままで，いくつかの話をくっつけてムリヤリひとつの結論を導き出している，といわざるをえないのです。

社説の要旨は次のとおりです。

1）　公共の場所で行儀の悪い若者は他人が目に入らない現代病患者では？

2）　免疫学，哲学，法学などにおいても，自己と他者の境界があいまいになっているという議論や現象がある（注）。

3）　電波メディアやパソコンネットワークなどの新しいメディアは，逆に人を孤立させ他者との接触をさえぎってもいる（注）。

4）　異質の人とつきあい，他者を発見しよう。そういう場を作ろう。

まず，1）の指摘は妥当なものといえるでしょう。歴史的に見てもあるいは世界的に見てもいろいろな面で新しい流れをたえず作ってきたはずの若い人たちが，自分のまわりが豊かである（あるいは豊かであると思いこんでいる）ので，他人のことや組織，システムの問題について話さなくなったりあまり興味をもたなくなってきています。このことは，少なくとも，おおざっぱな議論における比較の問題，あるいは統計の問題としては確かにそのとおりであるといえるでしょう。

次の2）もややこじつけ気味だという印象が多少残るものの，「現代病患者」の増加と同じ文脈で語るのもそれなりに面白いことだと思います（とくに，免疫学の多田富雄氏の話をほかで読みましたが，免疫レベルでの「自己」に関する話にはひきつけられました）。

さて，3）が問題です。最近のパソコン通信や計算機ネットワークなどの新しいメディア（これを総称して「電子ネットワーク」と以降では呼びます）の登場を想定しているのだと思いますが，それに関連して，「コミュニケーションの手段の飛躍的な発達が，人々に孤独をもたらす」と断定しているのです。

この主張の理由は，1）や2）を述べている部分には書かれていません。街で他人に迷惑をかける人とパソコン通信やネットワークを活用している人との相関があるなどという暴論は，さすがにどこでもされていないでしょう。

理由として考えられる部分は「自他の境界がぼやけていくことでもある。これを増幅するのがメディアだ」「密室にこもっても，パソコン一つあれば自在に外界に接触できる時代になってきた」「顔をつきあわせたり，身体を接しあったりしなくても，ものごとは動いていく」というあたりでしょうか。

はて，電話という新しいメディアが登場する前にくらべて，電話が登場してからのほうがちょっとでも孤独が増えたのでしょうか？　電子メールを使うことによって孤独が増えるような要因ができたのでしょうか？

もちろん，直接誰かになにかを話しに行く代わりにメールですませたのならば，そのことに限ってみれば，顔をつきあわす回数が少なくなるということは事実です。そして，それが孤独になんらかの形でつながるということもとりあえず認めることにしましょう，納得はしてませんが。

しかしですよ，電子ネットワークってただそれだけのものでしたっけ？　すぐ近所にいて面と向かって話せるような人と交歓しあうという使い方がメインなのでしょうかね？

いやいや，そんなことはないでしょう。電子ネットワークを使うことによって，それこそ従来では考えられなかったような人と人とのつながりが新しく生まれてきているというのがごく当然で客観的なものの見方ではないでしょうか？

ですから，4）はまるでおかしな主張に思えます。たとえば，3）と4）を変えて，次のようにしたらどうでしょうか。

3）　一方で，メディアが発達し，電子ネットワークでつながるようになってきて，さまざまな人と人の新しいネットワークが生まれてきた。

4）　新しい可能性をもった電子ネットワークを使って，異質の人とつきあい，他者を発見する場を作ることは，自己と他者の境界があいまいになっている現代において興味深いことではないか。

注：同社説は4つの節で構成されていましたので，本文中では4つの要旨に分けてあります。ただ，同社説内で掲載されている要旨では2）と3）が以下のようにひとつにまとまっていました。
「自己と他者の境界が薄れてきている。メディアの発達は，人を孤立させ，他者との接触をさえぎってもいる」

揚げ足を取りたくなる気持ち

「電子ネットワークを使うと差し引きして，孤独が増えるのか，人間同士の結びつきが増えるのか」などという不毛な議論は，ほ

とんど，「カメラを使うと魂を抜き取られるのか」レベルの話に思えてなりません。具体的なイメージがわかないとやっぱりそう思えてしまうのでしょうか？

具体的な例が，それこそ同じ朝刊の一面に載っています。「マルチメディアに自治体も積極姿勢」という記事です。その中には，「パソコン通信網に観光案内や害虫発生などの情報を流して意見を求めたり」とか，「静岡県に対する意見や質問を電子メイルで募集する事業を始めた」などの実例が挙がっています。

この記事は地方自治体レベルの話なので，個人と個人のコミュニケーションとは違うということもできそうですが，相手が県だろうと誰だろうとネットワーク上では自由に意見を出しあえるというところがこの新しいメディアの最大の特長ともいえます。また，基本的にこのような「まちおこし」運動は，社説のいうところの「異質の人とつきあい，他者を発見する」ための格好の場であるといえます。

ただし，この記事の中にもまた，多少気になる表現はあります。それは，「映像やデータ，音声などの情報を，パソコンやテレビなど電子端末を使って双方向でやりとりするのがマルチメディアだ」と高らかに定義しているところです。

これはどうなんでしょうかね？「双方向でやりとり」というのは，マルチメディアとは直接関係のない，また別の概念であるような気がするのですが……。まあ，それはそれでひとつの定義ということにしましょう。

それにしてもなぜ，社説でもって，「電子ネットワークなどの新しいメディアは孤独を増す」などと平気で断定するのか想像できません。『1984』というオーウェルの小説が書かれた時代からもう何十年たったというのでしょうか？

「若い人に他人に対する気配りがかけている」と怒るのはかまいません。もっともな面もあります。パソコン通信になじめなくて，若い人があっという間に情報を交換し

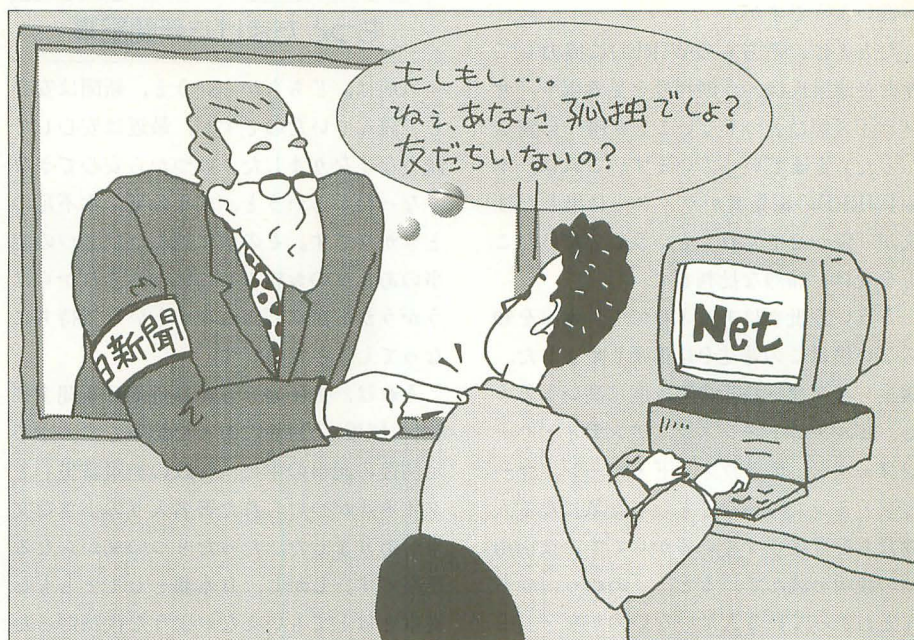



illustration : Haruhisa Yamada

ていることを嘆くのもいいでしょう。

しかし，パソコン通信が多くの人と人とのつながりを作ってきたという現実を無視して，逆に孤独を作るなどといっているのは理解できません。「これは現代の逆説である」などと書いていますが，逆説ではなく，「逆」そのものだと思います。

やはり，古いメディアは新しいメディアの出現に際して揚げ足を取りたくなるのでしょうか？　その気持ちはわかりますが，それならば，もう少し説得力のある方面から問題点を指摘するべきでしょう。僕としても気になる点はいくつもあります。

いくつか挙げるのならば，「若い優秀な人がインターネットに興味をもって長い時間を使っていろいろ遊んでいる（「ネット・サーフィン」）のだが，これはもしかして頭脳の浪費ではないのか？」とか，「ネットワーク上では自由に情報の発信者となりえるが，モラルの面に関して個人の良識にどこまで依存できるのか？」とか，「いわゆるネットワーク・ハイ（ネットワーク上で高揚しすぎて常軌を逸した状態になること）という状態は人間の精神のどのような状態を表しているのか，そして副作用や後遺症はないのか？」とかです。

WIRED

先の社説の締めくくりは「テレビやパソコンから離れて，時には，そんなことにも思いをめぐらせたい」となっていて，人と人とが線，あるいは電波でつながれている状態に対する嫌悪がひしひしと伝わってきます。

「人間同士，手を取り足を取り，酒でも飲みながら怒ったり笑ったりしながら，という状態こそがまともな人間の姿だ」というのでしょう。これには僕自身まったく賛成です。

しかし，人と人とをつなぐとさらに面白く未知の関係が生まれてくるものです。昨日までまったく知らなかった人でも，あるいは相手がどんな地位の人であっても基本的に対等です。電子の流れに情報を委ねると余計な情報は落とされるのです（ちょっと落としすぎではありますが）。

いみじくも，線でつながれた状態を意味する単語である『WIRED』という雑誌が出ています。2年前にアメリカで創刊されてカルト的人気を集めたといわれるこの雑誌の日本語版が出ました。ただし，残念ながら，元の雑誌にくらべて日本語版は評価

古いメディアは叫ぶ

が低いようですが。

たとえば，あちらのWIREDに協力してきた会津泉氏は，「創刊号を見る限り，サプライズがひとつもないことが唯一の驚きだった」とまでいっています。これは，本家WIREDの編集者がライターに出した注文が「驚かせてくれ」ということだったことを受けた痛烈な批判です。

そういう批判はあるとしても，本家を知らない僕はこの雑誌を結構楽しめました。編集レイアウト技術の完成度は高いと思うし，ビル・アトキンスン，アンディ・ハーツフェルド，デイヴィッド・バーン，マービン・ミンスキーなどといった気になる人物たちを登場させているからです。技術用語の説明を読んでいると，「あれっ，きちんとわかっている人はいないのかな？」と思うこともありますが，ちょこちょこっとした，ネットワーク関連や情報機器の情報は楽しめるものです。このようなイメージ先行型の雑誌もそれはそれで十分に楽しめるものです。

本家のWIREDは1カ月前の記事の内容をインターネットで公開していますので，登録して読めるようになりました。もちろん無料です。暇なときにじっくり読んでみようと思っています。

もっとたまげた新聞記事

以前は，どちらかというと，新聞は安心して読んでいたのですが，最近は安心して読めなくなりました。いつから安心できなくなったかというと，去年の夏の水不足のときからです。その頃に読んだひとつの記事のあまりのお粗末さに驚き，それからはうかうかと読んでられないという気持ちになってしまったのです。

それは，去年の8月20日付朝日新聞夕刊(名古屋版)の11面にあった記事です。水不足関連の記事の中に「行政の無策露呈」とタイトルがつけられた署名入りの小さな文章がありました。たった9つの文からなる記事です。しかし，日本語としてまともな文章とはとてもいえないような代物だったのです。たとえば，

「協議会が断水地域が拡大の方向を見せ，……」：主語「協議会が」がどこにもつながらない。

「土地改良区の理事長は，議員の○○氏や県会議員と政治家もいる。」：意味不明。

「こうした○○を恐れた××が恐れたためであることは間違いない」：少しばかり意味不明。

これ以外にも，日本語の「てにをは」がおかしいところがあったり，連続した2つの文中に「間違いない」が2個と「まちがいない」が1個含まれるというまったく日本語のセンスがないものだったり，と悲惨なものでした。このように，たった9つの文中にたくさんの文章的におかしいところがあるのです。意味以前の問題です。

さすがにこれにはあきれて，計算機上のアクシデントかなにかがあってこのような未推敲の文章が流れ出てしまったのかなと思い（われながらおせっかいかなとは思いましたが気まぐれを起こして)，問い合わせの手紙を出してみました。しかし，予想に反して反応はありませんでした。

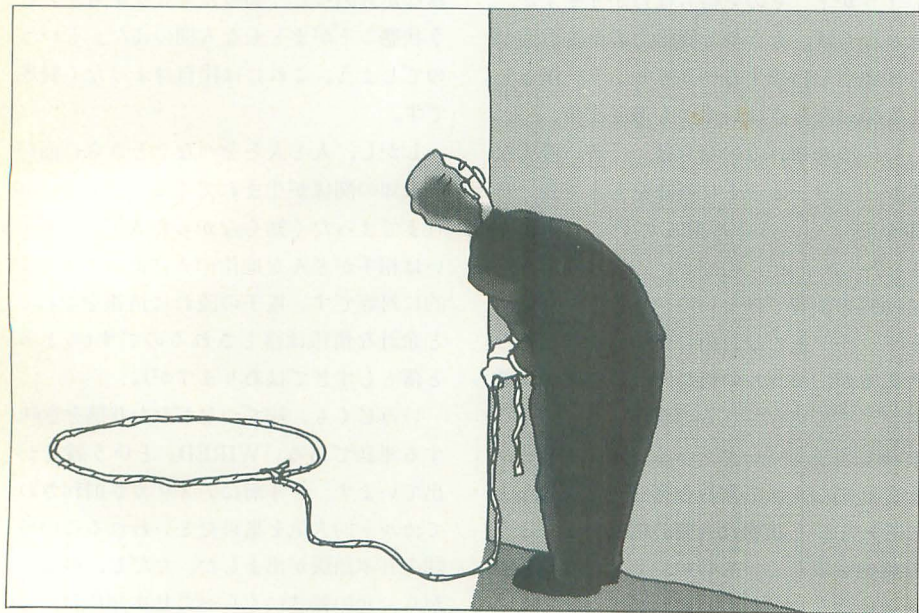
生まれてからずっと朝日新聞を読んでいるし当分読み続けるだろうとは思いますが，反応がなかったことで，ああこんなレベルだったのかとがっくりしてしまいました。まあ，この例は紹介した社説とは比較にもならない次元の低い話です。一度でも読み直せばこんなのは載らないはずですから(しかし署名記事だから謎です)。まあ，名古屋版だけの問題かもしれませんが。

ネットワークだからこそ

読者から電子メールをいただくと記事を書いているほうにとってもうれしいものです。

先週も，人工生命関連の大学院へ進みたいという相談とか，数カ月前の「このプログラムは生きているのか」で紹介した「Tierra」のプログラムについての問い合わせとかがありました。

阪神大震災に関する情報も電子ネットワークで飛び交っています。大学卒業以来一度も連絡のなかった人から昨日電子メールをもらいました。そして，たぶん新婚旅行のときに撮ったと思われる写真でしか顔を知らない編集の人に今回も電子メールでこの原稿を届けるのです。

e-mailアドレス
ari@info.human.nagoya-u.ac.jp
NIFTY-Serveから送信するには，上記のアドレス前にINET:をつける。

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートハガキの該当項目をすべてご記入のうえ，希望するプレゼント番号をハガキ右下のスペースにひとつ記入してお申し込みください。締め切りは3月18日の到着分までとします。当選者の発表は5月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞に当選できない場合がありますので，ご了承ください。

1 魔法大作戦

3名

X68000用

5″2HD版　9,800円（税別）

EAV　☎03(5410) 3100

グラフィックバリバリの縦スクロールシューティングゲーム。ゴブリガンを倒すまでひたすら撃ちまくるのだ(要X68000XVI以上のマシン)。

2 JOYCONT TURBO V

5名

スピタル産業　☎03(3251)2918

チタンブラック限定仕様の連射機能つきジョイパッドです。

3 A）岡村まんが祭 B）でんこちゃん☆パラダイス

各3名

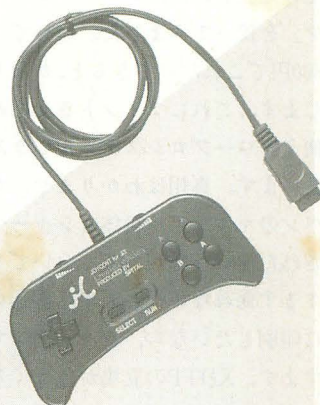
電子出版

冬コミで販売された同人誌を2種類，電子出版からいただきました。電子ちゃんワールドを満喫できる本です。

4 同人CD RISING EARTH

3名

MYU-RECORDINGS

「地球」をテーマにした同人CDです。全16曲で，(善)バビでお馴染み西川善司氏の曲も収録されています。

1月号モニタ当選者

A電話番号データベース「黒船」全国版　(高知県)多田　友

1月号プレゼント当選者

1スーパーストリートファイターII　(愛知県)山本英生　(茨城県)木下勝由　2　違法コピー防止啓蒙キャンペーンポスター(福岡県)蒲浦　修　田辺忠茂　(埼玉県)設楽　茂　(神奈川県)加藤政則　(栃木県)八木沢良二　(愛知県)宮下健輔　他5名　3MIYA-NET特製カレンダー　(奈良県)村井弘幸　(神奈川県)品川達郎　(大分県)後藤聡文　(富山県)松居啓樹　(静岡県)鈴木俊之　他5名　4ソフトバンク卓上カレンダー　(滋賀県)北畠　駿　(福岡県)小柳彰宏　(埼玉県)大塚俊宏　(東京都)木植幸男　(香川県)石田　源　(大阪府)福森　淳　(神奈川県)市川尚孝　(群馬県)久保田智久　(山形県)長尾浩敏　(北海道)横山宏一　他10名　(敬称略)

以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

短期集中 SX-WINDOWによるDTP

レイアウトを真似てみよう

Taki Yasushi 瀧 康史

前回はXDTPの概念について説明しました
今回は具体的な使い方に迫ります
基本部分が重要なのでしっかり理解してください

年末年始に8日間の休暇を取りました。実家に帰るとコンピュータがまともに触れない状態になるので，少しは気が楽になるかな……と思ったのですが，ううむ。触れないと触りたくなるものなんですねぇ。3日もすると文章のイメージが浮かんできて，メモ帳に残したくなってしまいました。そろそろ自分も「ライター」になってきたんだなあ……。

先日発売された,「シャーペンワープロパック」を使ってみました。いいですねぇ。6,800円でこれだけできると,かなり割安に感じます。これにフォントさえあれば，結構使えるワープロシステムができあがってしまいます。真相はわかりませんが，シャーペンのマニュアル自体もシャーペンで記述されたという噂があるくらいですから，ますます侮れない存在です。普通の人が普通に印刷したいなら，シャーペンで十分。ますます，XDTPの立場が危うくなってしまいそうですが，全体のレイアウトを考えなくてはならない印刷物を作るときなど，やはりXDTPじゃないと難しいでしょう。XDTPが，シャーペンに比べて有利な点は，この講座を読むことによって次第にわかることと思います。

用紙設定

それでは実際にXDTPを使ってなにか出力してみましょう。

つれづれなるままに，文章を書き連ねていって，途中で図を入れたりするシャーペンと違い，XDTPは一番最初に出力されるレイアウトのイメージを頭のなかでよく考えておく必要があります。シャーペンは，印刷する段階になって初めて「紙」を意識しますが，XDTPは最初から「紙」を意識しなくてはならないからです。したがって，最初にやることは用紙設定です(図1)。ここで，作る本の形を決定します。ここではOh!Xを真似てみて，左閉じの両面印刷，図1には出ていませんが，サイズはA4にしています。ノンブルはとりあえず，実際のデータ上の1ページ*1目を，印刷する1ページとし，ノンブルタイプは半角にします。あとの時刻，日付などは，入れるつもりはないので，そのままでいいでしょう。

*1 データ上のページは何ページを指定してもかまいません

マスターフォーム編集

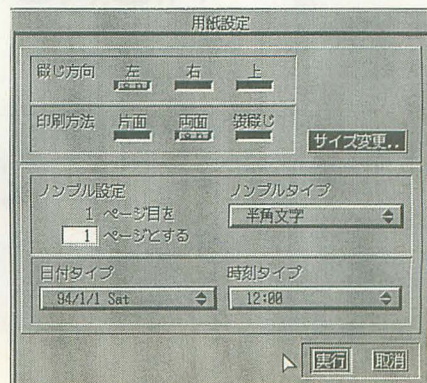
次はマスターフォームの編集です。これは，編集する全ページのレイアウトに影響しますから，基準となるものを偶数ページと奇数ページの2種類について決めておきます。ここでは本文を2段組みにし，下部に横線を引き，その横線の下に，偶数ページなら左下，奇数ページなら右下にノンブルを入れることにしましょう(図2)。

まず，ノンブルを入れる位置にテキストフレームを置きます。このテキストフレームですが，まだ文の全体量がはっきりしていないならば少し大きめにとっておくといいでしょう。さて，ノンブルの設定ですが，ツールウィンドウ(図3)でテキストをクリックし，ノンブルを入れたいテキストフレームをクリックします。それから，テキストメニューの「特殊文字入力」を選びます。ノンブルの位置は奇数ページではページの右端にくるので行揃えを右寄せに，偶数ページでは左端にきますから行揃えを左寄せにしておきます。ただ，なぜか右寄せの場合，ノンブルのあとに1文字分のスペースを置かないと，ノンブルの書体を斜体にしたときに，傾いた右肩がはみ出てしまいますので注意してください。

ノンブルのフォントは，Amadeusを使うことにしましょう。文字サイズは8ptぐらいにします。Amadeusを利用すると，文字設定がいかにせよ斜体になるので，カーニングをするほうがよいでしょう。カーニングしないと図4-1のように文字幅が絶対的になってしまいますが，カーニングをすると図4-2のように文字スペースを都合よく合わせてくれます。

偶数ページのノンブルにはページ番号と全ページ共通の誌名を，奇数ページにはページ番号と記事の名前を入れることにします。ただし記事名は，本全体で同じではないですから，記事名を書くためのテキストフレームだけを置いておきます。

ページ単体で見れば，本文は左の段から右の段へ移っていきますから，この2つだけはマスターフォームの上でリンクしておきます。また，本の閉じ方を考え，奇数ページは全体を右側に少し寄せ，偶数ページは全体を少し左側に寄せたほうがよいかも

図1 用紙設定のダイアログ

しれません。

ここまでできたら，ツールウィンドウでポインタをクリックし，ノンブルの書かれているテキストフレームをダブルクリックします。そうすると，このテキストフレームの環境設定が行えます(図5)。

ここでテキストフレームを「回り込みさせる」に設定すると，別のテキストフレームが近くにきたときに，その中のテキストが回り込んでしまいます。ノンブルのせいでほかのテキストフレームの内容が回り込んでしまっては困るので，回り込み設定にはしません。またこのノンブルの書かれたテキストフレーム自身，本来回り込みさせるように設定してあるフレームの近くにいったり重なったりしたときでも，回り込みしないほうがよいので，回り込みの設定は一番左の「上書き」に設定します。回り込みをしない(させない)設定が優先されるので，これでOKです。図6にいろいろな回り込み設定をしたものを載せておきますが，非常にわかりにくいので自分で試してみるのがよいでしょう。

どのページにもほぼ共通の，基本となるレイアウトの内容はこのぐらいですから，マスターフォームの編集はこれで終わりです。

ドキュメント編集

●テキストフレーム

マスターフォームが完成したら，ドキュメント編集に戻します。マスターフォーム編集では，ページ番号は「P」と表記されていましたが，ドキュメント編集モードでは，ちゃんとページ番号が出ているはずです。フォントは設定したとおりにAmadeusになっています（写真1）。

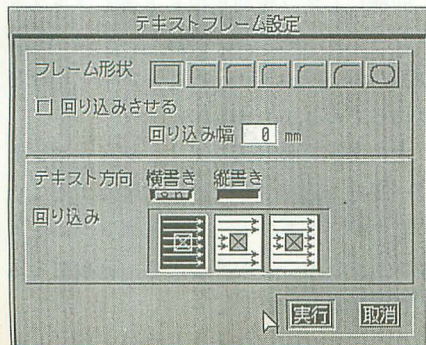
図5　テキストフレーム設定のダイアログ

とりあえず，1ページ目を編集することにします。ところで，このまま印刷するとどうなるでしょうか？　実際にやってみればわかります。結果は，下のほうにある横線と，ページ番号のみ印刷されるはずです。XDTPはWYSIWYGなツールですが，デフォルトの状態では，ドキュメント編集のときに見える状態と印刷結果が違います。ただ，制限はありますがちゃんとWYSIW

写真1　自動的にページをふってくれる

図2　今回作成するマスターフォーム

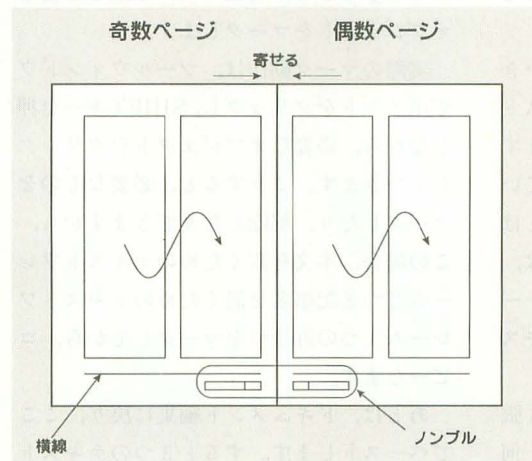



図4　カーニングの一例

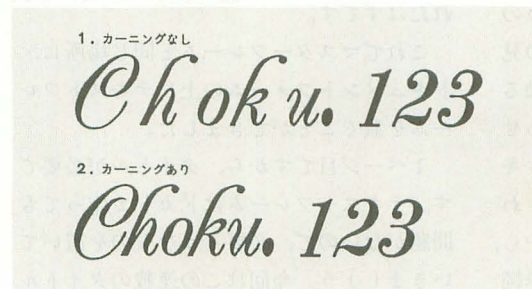



図3　ツールウィンドウ

図6　いろんな回り込み設定

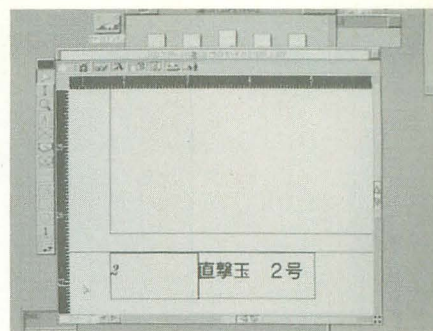
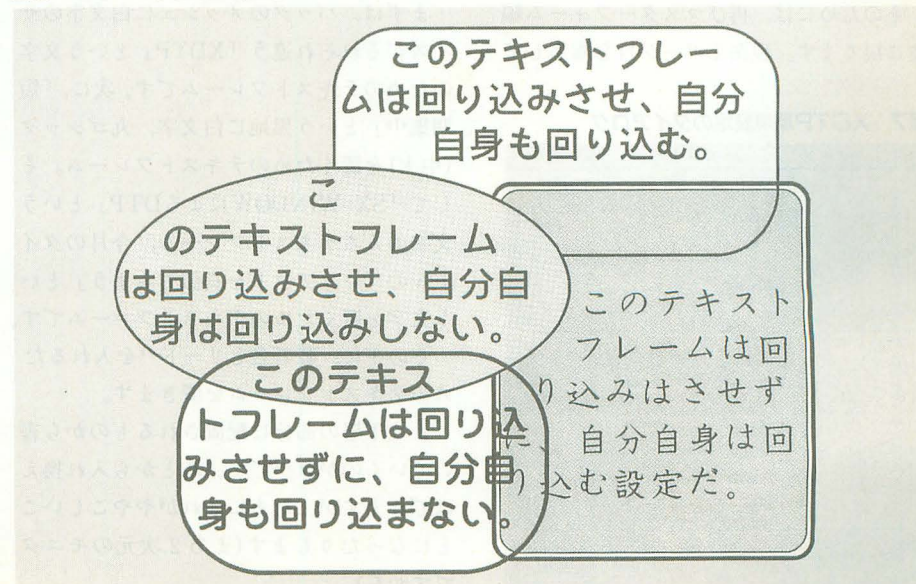
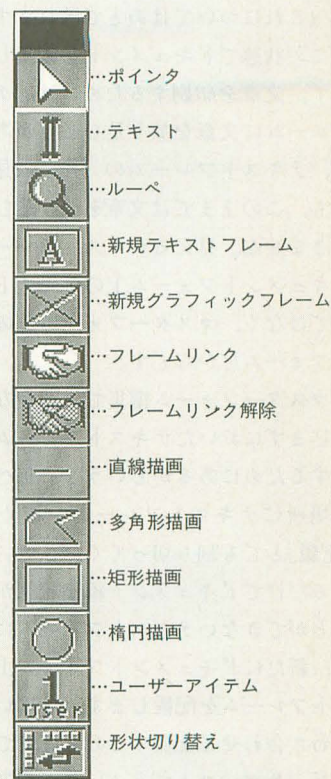
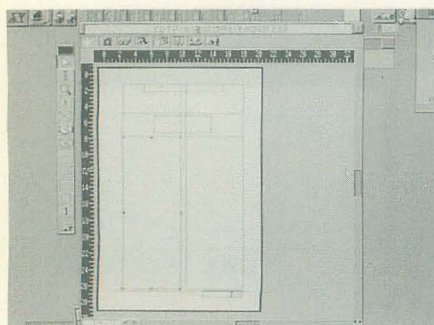



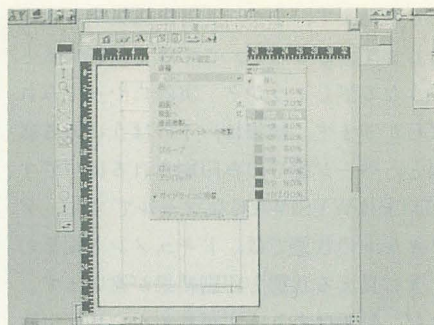
写真2　マスターフォームのフレームが見える

写真3　塗りつぶしを30%に設定

*YG*にするモードもありますので，御安心を（これについてはあとで触れます）。

この状態でドキュメントを編集していきます。文章を印刷するためには，テキストフレームに文章を放り込む必要がありますが，テキストフレームの「枠」は見えていても，このままでは文章を放り込むことはできません。見えるテキストフレームは，ドキュメントフォーム上のテキストフレームではなく，マスターフォーム上のテキストフォームだからです。

マスターフォーム編集で文章をなにも張り込まずにおいたテキストフレームは，何をするためにあるかというと，毎ページ同じ場所にテキストフレームを置くための「定規」とでも割り切ってください。この見えるだけでドキュメント編集状態から触ることができないテキストフレームに合わせて，新たにドキュメントフォーム上にテキストフレームを配置します。とはいえ，わざわざ合わせて配置するのも面倒ですから，コピー&ペーストでうまいこと作業を単純化します。

そのためには，再びマスターフォーム編集に戻ります。現在1ページ目を編集しているのですから，奇数ページの上で必要なオブジェクトをマークします。

実際のマーク動作は，ツールウィンドウでポイントをクリックし，SHIFTキーを押しながら，必要なオブジェクトをクリックしていきます。こうすると，必要なものをマークしたり，解除したりできますから，この場合，本文を置くためのテキストフレーム2つと記事名を置くためのテキストフレーム1つの計3つをマークしてから，コピーします。

あとは，ドキュメント編集に戻り，ここでペーストします。すると3つのテキストフレームがドキュメントフォーム上に置かれたはずです。

これでマスターフレームと同じ場所に，ドキュメントフォームの上にテキストフレームを置くことができました。

1ページ目ですから，タイトルが必要です。テキストフレームはドカドカ作っても問題がないので，必要そうなものを置いていきましょう。今回はこの連載のタイトル部分をイメージして作っていきます。

まずは，バックのメッシュに白文字のサイズがそれぞれ違う「XDTP」という文字のためのテキストフレームです。次に，「短期集中」という黒地に白文字，丸ゴシック(中太)を置くためのテキストフレーム。そして「SX-WINDOWによるDTP」という文字の入るテキストフレーム，今月のタイトルの「レイアウトを真似てみよう」という文字を置くためのテキストフレームです。

その下に，著者名とリード[*2]を入れるためのテキストフレームを置きます。

より下層の部分に配置されるものから書いていくのがコツです。あとから入れ換えはできますが，たまにこれがややこしいことになったりします(まあ2次元のモニタですから)。

このようなテキストフレームを置く作業は，ディスプレイの解像度が高く，ドット比が1：1でないと，なかなか仕上がりが想像できません。なお，どうしてもテキストフレームが1ドット単位で合わないときには，環境メニューの「XDTP環境設定」で「ガイドライン吸着」を「しない」に設定すると，細かく設定ができるようになります（図7）。

図7　XDTP環境設定のダイアログ

まあ，文章を張り込んでからもテキストフレームのリサイズや移動はできますから，最初はおおざっぱな配置でもいいかもしれませんが。

ここまでできたら，本文を入れるための，テキストフレームを小さくします(写真2)。そうしても，元のサイズのテキストフレームは残っているように見えますが，これはマスターフォーム上のものなので無視してかまいません。

各テキストフレームの環境も設定しなくてはなりません。これはテキストフレームをポインタ指定した状態でダブルクリックです。すべてのテキストフレームは，「回り込みさせない」設定にします。自分自身の回り込みの設定は，本文が入るテキストフレームを除いて，「回り込みしない」設定にします(一番左)。本文はひょっとしたら「図」が入って回り込みしてほしいときがあるかもしれないので，「回り込みする」設定にしておきます。

では文章を張り込みましょう。

まず，背景の「XDTP」という文字です。このテキストフレームは，多少濃いめのメッシュがかかります。テキストフレームをセレクトした状態で，オブジェクトメニューの「塗りつぶし」をベタ30%にします。モニタ上よりも印刷結果のほうが，「濃く」なるので，少し薄めを選ぶべきです(写真3)。

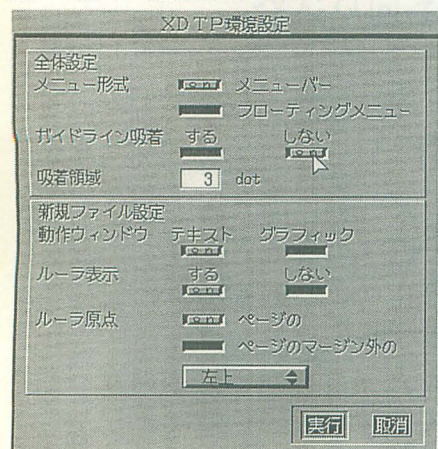
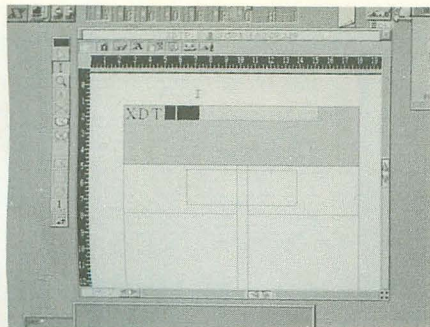
同様にして「短期集中」と書くテキストフレームを100%(真っ黒)に，「SX-WINDOWに～」というテキストフレームを20%にしておきましょう。

次に文字を張り込みます。シャーペン上で一度打ち込んでからペーストしてもよいのですが，まあひとつのテキストフレームにつき，ひとこと程度なので，直接打ち込んでも問題ないでしょう。

これにはツールウィンドウのテキストを

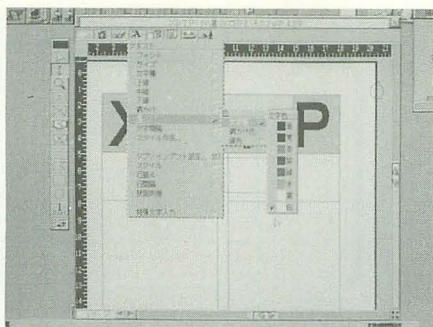
写真4　フレーム内に文字を入力する

写真5　テキストメニューで文字色を決定

写真6　グラフィックモードでの表示

クリックして，背景の「XDTP」と書くつもりのテキストフレームをクリックします。するとテキストフレームの中でカーソルが点滅するので，「XDTP」と打ち込み，確定します（写真4）。

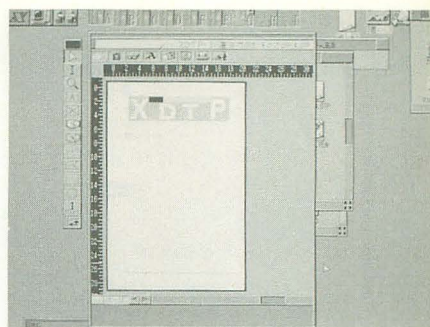
そして，入力した文字列をマウスでマークし，フォントを極太角ゴシックあたりに設定します。文字サイズは1文字ずつ違うので，1文字ずつマークします。「X」はいっぱいのサイズになるように文字サイズ設定の「その他」で設定します。入らないサイズだと，テキストフレームの左側に「×」マークが出るので，小さくしてください。「D」と「T」は縦に潰れているので，水平比率を変えなくてはいけません。例では，「D」と「T」は75ptの水平比率115％，「X」と「P」が96ptの水平比率100％にしています。本来，XDTPと文字を書いた時点で，文字間隔を均等割り付けにすればよいはずなのですが，どうもうまくいかないので，適当なところにスペースを入れて誤魔化しています。このあたりは，バグに近い仕様なのでなんとか直してほしいところですね。おそらく，文字間を開けるときに，文字の左に文字間をとって，それをひとつのオブジェクトとしているため不具合が起きているのではないでしょうか？

また，上の2つのテキストフレームが，「X」と「P」に重なるようならば，リサイズしてください。

さて，「XDTP」という文字は，白文字ですから，色を白に変えなくてはなりません。「文字種で白抜きか？」と考えた人は，鋭いですが外れです。オブジェクトメニューの「色」設定で「白」にするだけ……ですが，写真5で見ると確かに白と設定してあっても，画面中の文字色は白になっていません。実際に印刷すると，白文字になります。これが，実際の印刷と画面が違う，2つ目のギャップです（1つ目はテキストフレームの線）。印刷される状態を画面で確かめるには，ドキュメント環境を「テキスト」から「グラフィック」に設定する必要があります。もしも，あなたのSX-WINDOWの環境が「6万色」モードであるなら，GRWを開きます。16色モードであるなら，そのままでかまいません[*3]。

ここまで作ったページを6万色モードのグラフィックモードで見たのが写真6で，16色モードのそれで見た場合も，写真6とほぼ同じです。どうですか？　グラフィックモードでは，テキストフレームの線も消えて，紙に印刷したのと同じようなイメージになっているでしょう。

ただし，6万色のグラフィックモードは，画面が512×512のGRWの中でしか編集できないので，ページ全体を適度な大きさで一度に見ることができません。また，16色モードのそれでは，テキストモードと同じ広さの画面を使えますが，カラーイラストなどを張り込んだときに，色がきちんと出ません。どちらも一長一短ですが，XDTPに関していえば，6万色モードより16色モードのほうが都合がよいような気がします。

それから，実際の編集ではグラフィックモードとテキストモードをいったりきたりするので，ショートカットキーを覚えておくといいでしょう。OPT.1＋Rでグラフィックモード，OPT.1＋Eでテキストモードになります。6万色モードでは，GRWがないとダイアログが出て怒られてしまうので注意してください。また，残念ながらSX-WINDOWの都合上，再起動なしには16色モードと6万色モードをいったりきたりできません。

タイトル部分などの残り4つのテキストフレームは適当に編集してみました。さしたる問題はないでしょうが，テキストのデフォルトのサイズが24ptと，なぜかむちゃくちゃ大きいため，小さなテキストフレームに文字を打ち込むと，いきなり「×」マークになってしまいます。こうなると直接マークできませんから，テキストフレームの中をテキスト入力モードのカーソルの状態でクワドラプルクリック[*4]すれば，文字が全選択になります。全選択できれば，フォントサイズを変えることは簡単にできるので，あまり問題ないでしょう。

また，黒地のときはグラフィックモードでしか文字が見えなくなってしまいますので注意してください。

本文を埋め込むのは簡単です。まずは，シャーペンなどで作成した文章をコピーしておきます。それからツールウィンドウのテキストをクリックし，放り込みたいテキストフレームをクリックすれば，カーソルが点滅するのでペーストしてください。

放り込んだあと文章を全選択します。テキストフレームの文字サイズのデフォルトは，24ptですから，これを8ptぐらいに変更しましょう。また，フォントは明朝体ベジエあたりを適当に使います。半角フォントは，明朝体ベジエならifm.envの中の設定次

SX-WINDOWでのイメージ印刷

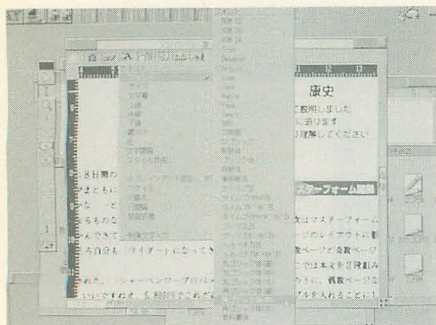
愛知県の笹山和宏さんよりハガキをいただきました。

「SX-WINDOWでイメージを印刷すると汚くなってしまうのはシステムリソースのバグが原因です。IVM.Xではありません。このバグの出ない自作の変換ルーチンが偶然「UNIX USER」の付録CD-ROMに収録されたので機会があれば一度お試しください。1994年11月号LibCD Vol. 8 の¥net news¥fj¥binaries¥wisc¥にあります」

ということで，今回の印刷例ではこれを使用させていただきました。どうもありがとうございます。ただ，これを利用するには内部リソースを直接いじるのでかなりの知識が要求されますのでご注意ください。あと，このリソースの解析データも送ってくだされば助かるんですけど，ダメですかねぇ。

写真7　文字のフォントを変更する

写真8　オブジェクトメニューで線種を決定

写真9　グラフィックフレームを配置

第で，フォントをタイムズなどに設定することもできます。デフォルトは禁則なしですから，禁則をするのを忘れないようにしましょう。私は，追い込み禁則をよく利用します。

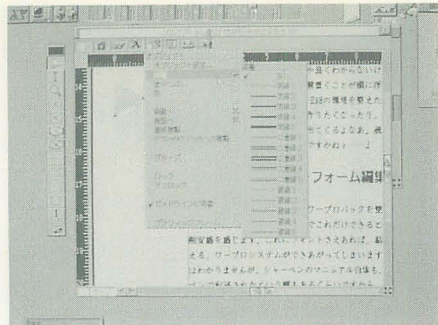
途中の「用紙設定」や「マスターフォーム編集」は小見出しですから，文字サイズを12ptにして，フォントを角ゴシック中太ぐらいに設定しておきましょう(写真7)。これで，本文のテキストフレーム編集は終わりです。

ノンブルの隣にあるテキストフレームには「SX-WINDOWによるDTP」という文字を入れます。

テキストフレームだけでも，使い方次第でかなりのことができてしまいます。

●グラフィックフレーム

それでは，多角形描画とグラフィックフレームの利用について説明します。どちらもオブジェクトとしての概念は，テキストフレームと一緒ですから，ほとんど同じ操作でものを置くことができます。

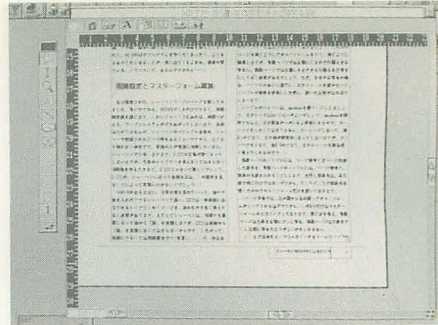
まずツールウィンドウの多角形描画をクリックし，小見出しの背面に置く三角形を書いてみます。3つの頂点を適当でいいですから決め，それを閉じて終了します。この三角形自体オブジェクトですから，オブジェクトメニューの「塗りつぶし」「線種」の設定がそのままテキストフレームのときと同じように利用できます。

ここでは，線種はなし，塗りつぶしは30%に設定しておきます(写真8)。配置が文章の上にきますが，取りあえずは気にしないで，ちょうどいい場所に置いておきます。3つの頂点の小さな四角形のマークをつまめば形は変えられるので，ほどよく変形をしてそれらしいところに置きます。

そして，この三角形がセレクトされた状態で，オブジェクトメニューの「背面へ」という項目を選びます。こうすると，順序が入れ替わります。

次はグラフィックフレームの扱いです。グラフィックフレームはイメージ画面を張ることができるフレームです。あくまでオブジェクトですから，実際の使い方は，ほかのものとほとんど変わりません。

右下に図を入れることにしましょう。印刷例の図1は用紙設定のダイアログなので，これを画像ファイルとしてもってくる必要があります。これには「SX-WINDOW開発キット用ツール集」の中に入っている，「カメラ」を利用します。

カメラは普通，GLM形式の無圧縮で保存しますが，純正のGLMには不具合があり，最大サイズ＄40000を超えるデータの保存に失敗します。これを修正するパッチがPCVANのXICLUBなどにアップされているので，これもあると大きなデータも大丈夫です。

カメラはダイアログにも効きますが，実際の保存はダイアログ終了後に行われます。こうしてできたファイルを，キャンバスで見ると，たいてい，余計なところがあるので，グラフィックツールで加工します。残念ながらこれは，SX-WINDOWを終了して，MATIERなどを利用するしか，現状では手立てがありません。

MATIERはGLM形式をサポートしているので，これで部分セーブを行い，SX-WINDOWを再起動し，キャンバスで確実にセーブされているかを確認します。思ったとおりにセーブされているなら，XDTPを起動し，グラフィックフレームを置きましょう。

まず，ツールウィンドウの新規グラフィックフレームをクリックし，グラフィックを張り込みたい場所に置きます(写真9)。グラフィックツールなどで矩形を書く要領です。このサイズは，ドット比が1：1のディスプレイモードでないと，実際の印刷と異なるので注意してください。

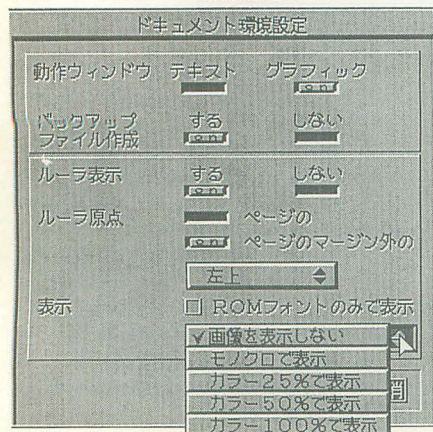
このグラフィックフレームはあくまでオブジェクトですから，これをダブルクリックし，オブジェクトの設定を行います。やり方はテキストフレームのときと同じです。回り込みさせる設定で，適当に幅を2㎜ぐらいに設定しておきましょう。

この大きな「×」のついたグラフィックフレームに，先ほどセーブしたファイルをドラッグして放り込みます。もしも，ここでウィンドウが「／」になってしまったなら，それはメモリが確保できないことを意味しています。放り込む画像のサイズの分だけは最低でもプロセス情報で連続空きメモリとして開いていないといけません。

また，放り込んだとしても，多分，画像は乱れています。この画像の解像度は，環境メニューの「ドキュメント環境設定」によって設定します(図8)。画像の表示品位を上げるほど，メモリとCPUパワーが必要になることはいうまでもありません。

あとは，図のタイトルです。

タイトルはテキストですから，適当な大きさでテキストフレームを置きます。この

図8　ドキュメント環境設定

テキストフレームは回り込みさせる設定にし，テキスト自体は回り込まない設定にしないといけません。

これで終わりです。

グラフィックフレームを利用してイメージを張り込むと，メモリを湯水のように使い始め，全体速度がどっと遅くなり，さらには，印刷も涙が出るほど遅くなります。セーブも起動もぐっと重くなります。

これらの原因のほとんどは，XDTPではなく，SX-WINDOWのシステム上の都合によるものです。

＊2 本文の内容を要約したもの
＊3 両者の設定は，コントロールパネルの「画面モード」で行います
＊4 連続4回左クリックすること

最後に

今月は，実際にページを作成しながら，その手順とポイントを解説しました。多少重複する部分もあったかと思いますが，実際にXDTPを触りながら読んでいただければ，いっそう理解は進むと思います。

まだ持っていない人のなかには「私にはXDTPはいらないかな」と結論づけた人もいるでしょう。しかし，印刷も凝ってみると，なかなか面白いものですよ。

ところで，今回，同人誌を作る際に感じた教訓をいくつか挙げておきます。

いくら自分の家ですべてできるからとはいえ，

1） 台割りはきちんと書きましょう

台割りというのは，何ページから何ページまでにどの記事がくるのかという予定表のようなものです。これがないと，ほかのページの編集が終了していても，ノンブルがつけられないので，印刷できません。印刷時間はイメージが大きければ大きいほど，ひたすらかかります。A4タイトルモノクロ1ページをX68030で印刷するのに，5時間以上かかったという記録もあります。

2） 役割分担をしましょう

XDTPはマルチタスクではありますが，印刷中はなにもできないほど重くなります。ひとつのマシンをワークマシンにしていると，印刷待ちをしなくてはなりません。同人誌をXDTPでつくるなら，XDTPを使える環境と人間を少なくても3台(人)ぐらいは用意しておくべきです。まあ，そういう状態を想定してXDTPを人に教える手助けにするためにも，今月の原稿を少しやさしめに書いたのですが……。

3） ファイルは分割しましょう

XDTPのマニュアルには，あたかも通しで100ページも200ページもの本が印刷できそうに書いてありますが，そんなことはありません。やっぱり，ひとつの記事ごとにファイルを分けるべきです。ちゃんと最初のページをNページに設定する，というのがあるんですから……。

これは，役割分担にも影響しますし。

今月の教訓はこんなところです。もうちょっと深い使い方をすると，また違った教訓がいえますが，今月使用したレベルではこのぐらいでしょう。

もっとも，フォントが100Mバイト，ハードディスクが1Gバイト，マシンはX68030，メモリは12Mバイト，レーザープリンタつき……なんてシステムをサークル内の人間がたくさん持っているとは思えませんが。

そのあたりは，工夫してなんとかしましょう。

印字例(LP-1500で出力，66%に縮小)

短期集中 SX-WINDOWによるDTP

XDTP レイアウトを真似てみよう

Taki Yasushi 瀧 康史

前回はXDTPの概念について説明しました
今回は具体的な使い方に迫ります
基本部分が重要なのでしっかり理解してください

年末年始に8日間の休暇を取りました。実家に帰るとコンピュータがまともに触れない状態になるので，少しは気が楽になるかな……と思ったのですが，ううむ。触れないと触りたくなるものなんですねぇ。3日もすると文章のイメージが浮かんできて，メモ帳に残したくなってしまいました。そろそろ自分も「ライター」になってきたんだなあ……。

先日発売された，「シャーペンワープロパック」を使ってみました。いいですねえ。6,800円でこれだけできると，かなり割安に感じます。これにフォントさえあれば，結構使える，ワープロシステムができあがってしまいます。真相はわかりませんが，シャーペンのマニュアル自体もシャーペンで記述されたという噂があるくらいですから，ますます侮れない存在です。普通の人が普通に印刷したいなら，シャーペンで十分。ますます，XDTPの立場が危うくなってしまいそうですが，全体のレイアウトを考えなくてはならない印刷物を作るときなど，XDTPじゃないと難しいでしょう。XDTPが，シャーペンに比べて有利な点は，この講座を見ることによって次第にわかることと思います。

用紙設定

それでは実際にXDTPを使ってなにか出力してみましょう。
つれづれなるままに，文章を書き連ねていって，途中で図を入れたりするシャーペンと違い，XDTPは一番最初に出力されるレイアウトのイメージを頭のなかでよく考えておく必要があります。シャーペンは，印刷する段階になって始めて「紙」を意識しますが，XDTPは最初から「紙」を意識しなくてはならないからです。従って，最初にやることは用紙設定です(図1)。ここで，作る本の形を決定します。ここではOh!Xを真似てみて，左閉じの両面印刷，図1には出ていませんが，サイズはA4にしています。ノンブルはとりあえず，実際のデータ上の1ページ*1を印刷する1ページとし，ノンブルタイプは半角にします。あとの時刻，日付などは，入れるつもりもないので，そのままでいいでしょう。

*1 データ上のページは何ページを指定してもかまいません。

マスターフォーム編集

次はマスターフォームの編集です。これは，編集する全ページのレイアウトに影響しますから，基準となるものを偶数ページと奇数ページの2種類について決めておきます。ここでは本文を2段組みにし，下部に横線を引き，その横線の下に，偶数ページなら左下，奇数ページなら右下にノンブルを入れることにしましょう(図2)。

まず，ノンブルを入れる位置にテキストフレームを置きます。このテキストフレームですが，まだ文の全体量がはっきりしていないならば少し大きめにとっておくといいでしょう。さて，ノンブルの設定ですが，ツールウィンドウ(図3)でテキストをクリックし，ノンブルを入れたいテキストフレームをクリックします。それから，テキストメニューの「特殊文字入力」を選びます。ノンブルの位置は奇数ページではページの右端にくるので行揃えを右寄せに，偶数ページでは左端にきますから行揃えを左寄せにしておきます。ただ，なぜか右寄せの場合，ノンブルのあとに1文字分のスペースを置かないと，ノンブルの書体を斜体にしたときに，傾いた右肩がはみ出てしまいますので注意してください。

ノンブルのフォントは，Amadeusを使うことにしましょう。文字サイズは8ptぐらいにします。Amadeusを利用する以上，文字設定がいかにせよ斜体になるので，カーニン

図1 用紙設定のダイアログ

SX-WINDOWによるDTP 1

another
CG
world
vol.42
28 JAN '95
© KYOKO EGUCHI

1

2

3

mmm
m
mm
m
m
mm
m
mm
4

5

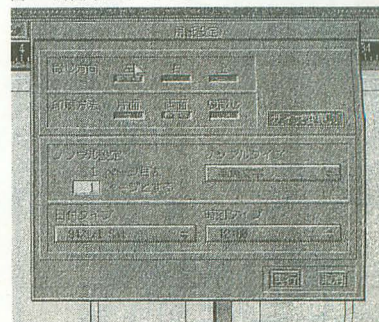
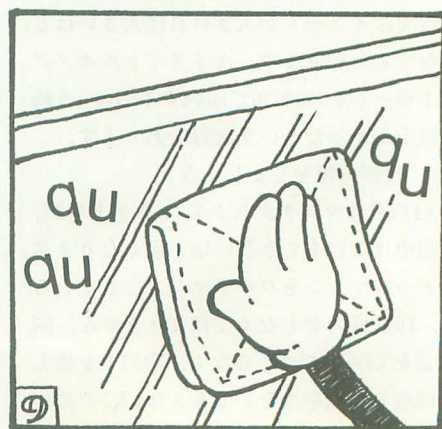
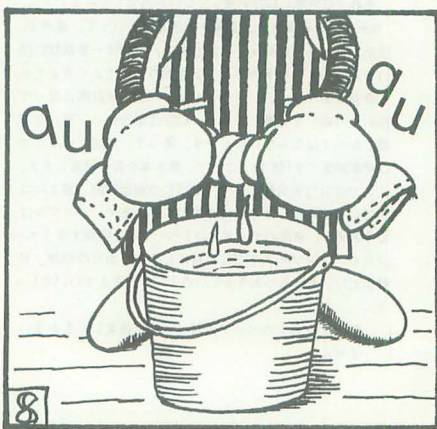
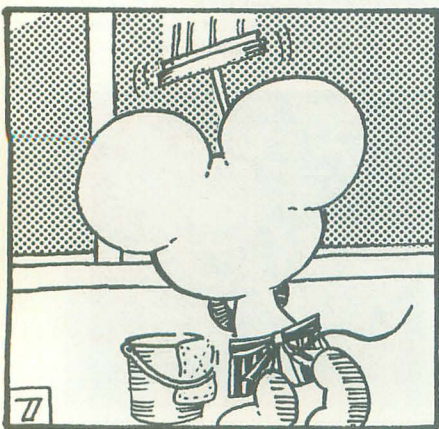
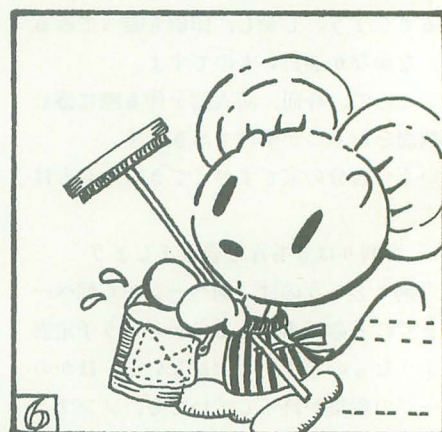
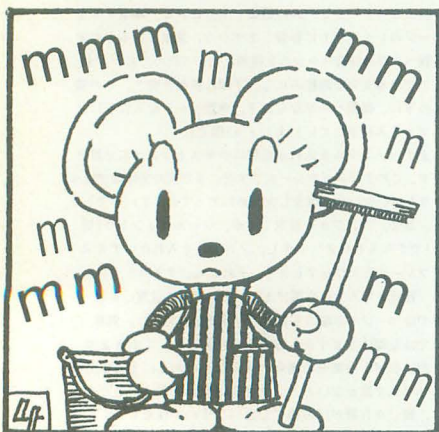
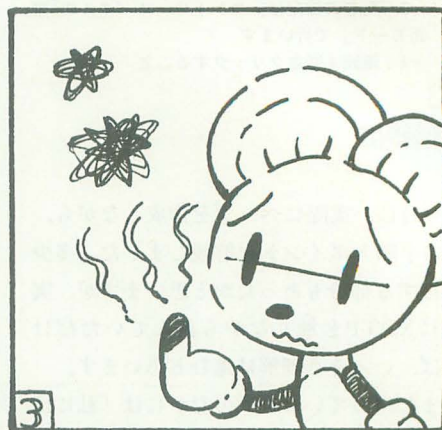
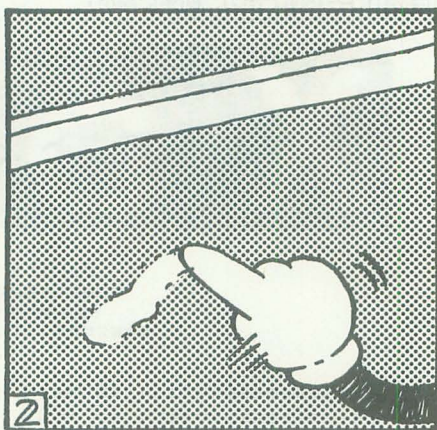
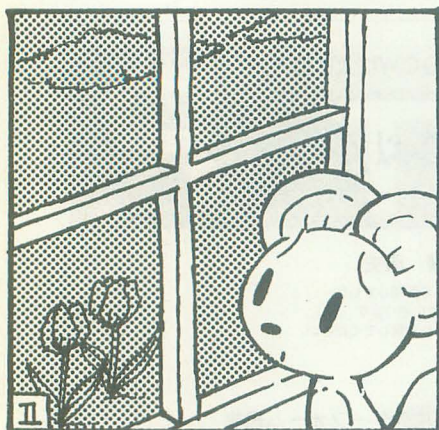
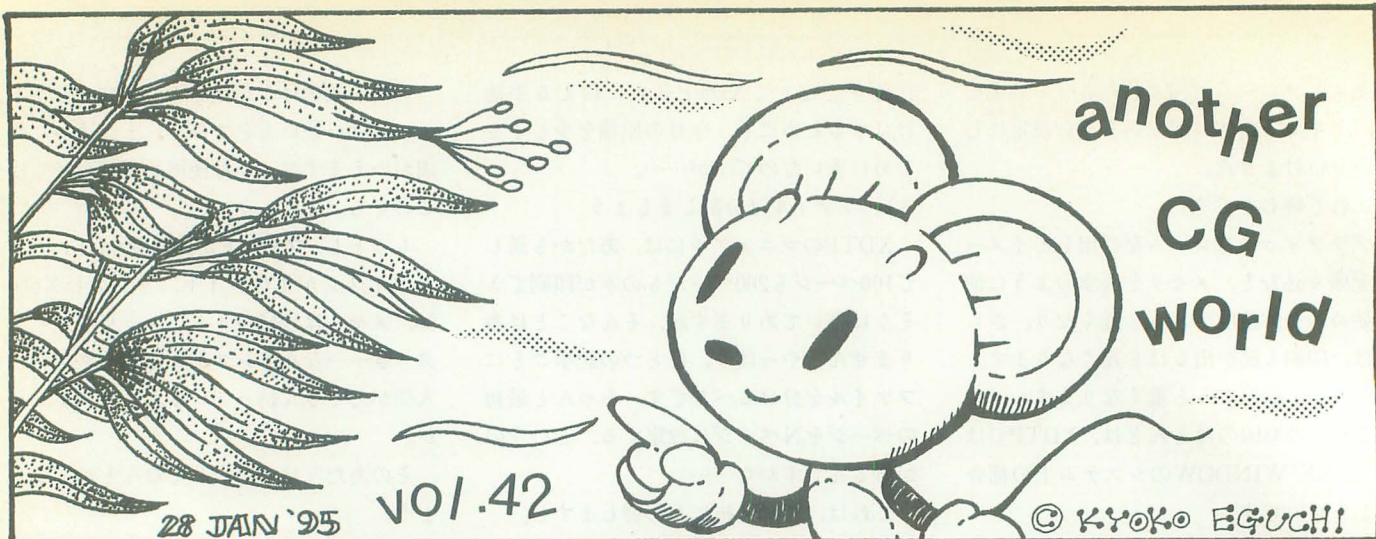
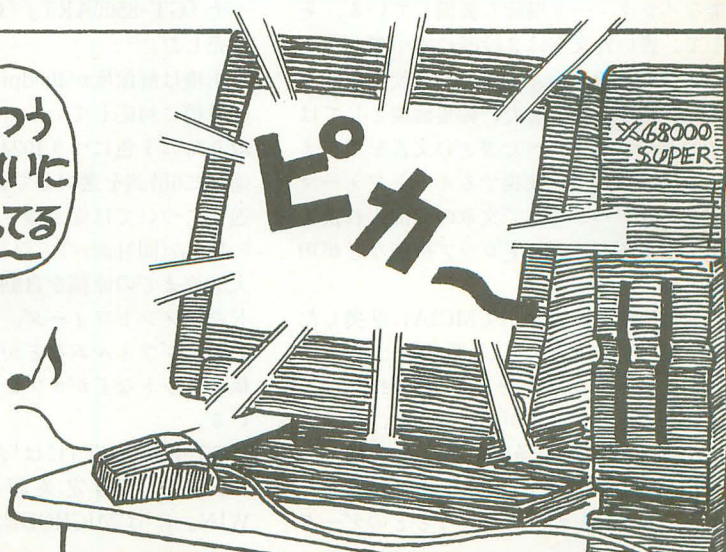
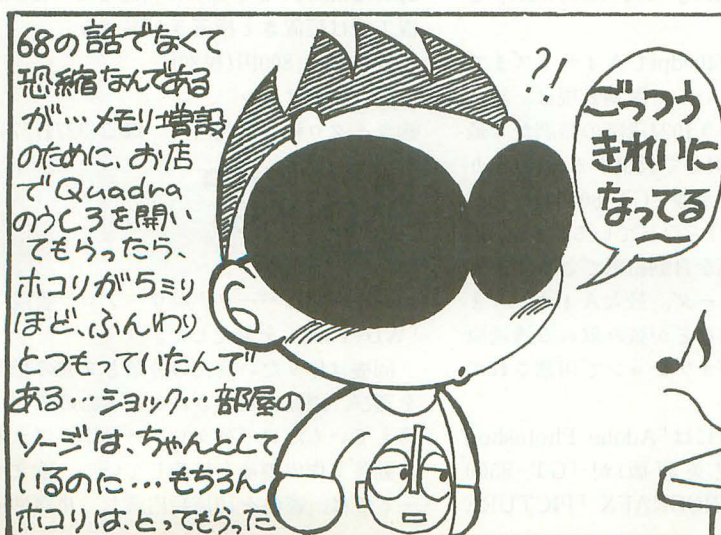
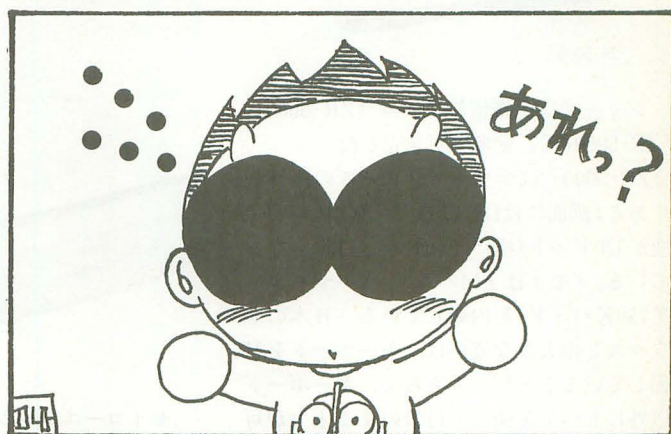
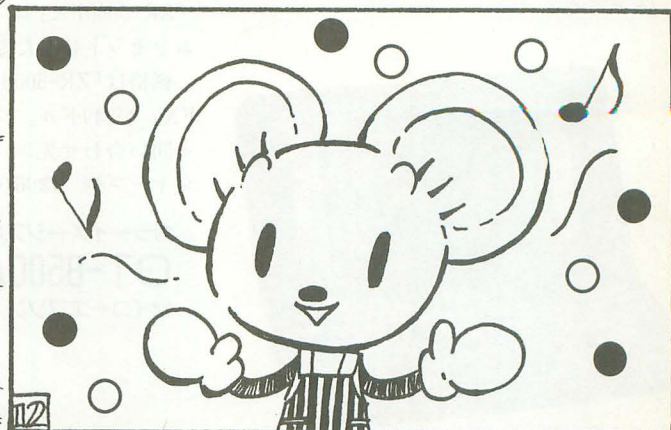
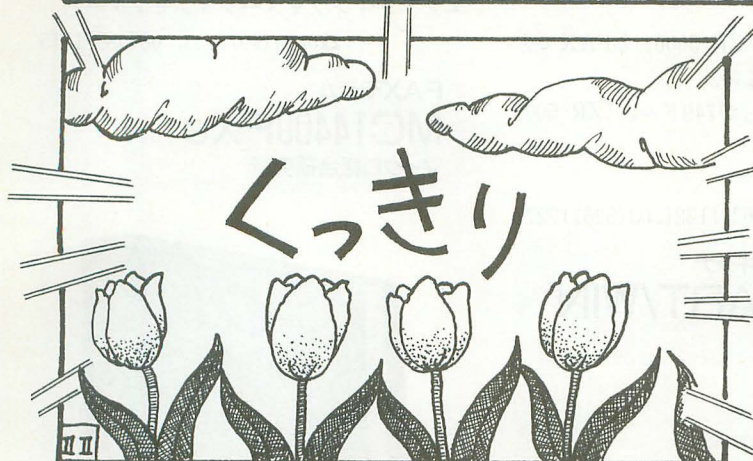
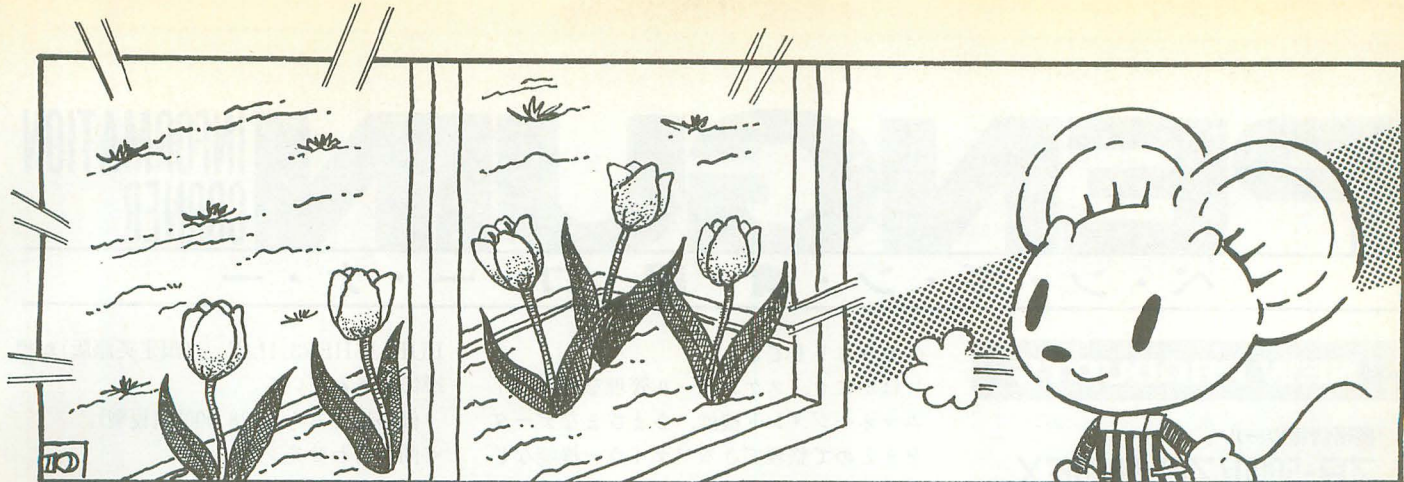
6

7

qu
qu
8

qu
qu
qu
9

くっきり
さーてと CGでも 作るかいね〜
あれっ?
68の話でなくて恐縮なんであるが…メモリ増設のために、お店でQuadraのうしろを開いてもらったら、ホコリが5ミリほど、ふんわりとつもっていたんである…ショック…部屋のソージは、ちゃんとしているのに…もちろんホコリは、とってもらった
?!
ごっつう きれいに なってる〜
ピカ〜
X68000 SUPER

PENGUIN INFORMATION CORNER
ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

携帯情報ツール ZR-5000/ZR-5000FX シャープ

ZR-5000

シャープは携帯情報ツール「ZR-5000」「ZR-5000FX」を米国で発売した。

「ZR-5000」はザウルスシリーズの海外版である。画面にはDFSTN液晶を用い，解像度が320ドット(横)×240ドット(縦)となっている。メモリは1Mバイト(ユーザーエリア750Kバイト)を内蔵している。日本のザウルスとの大きな違いは，キーボードを装備していることだ。もちろん，キーボード以外にもペンを使って自筆のまま入力が可能なインクノート機能も装備している。そして，書いた文字は3段階に縮小でき，それを参照しながらキーボードを使って別文書を作成できる。また，編集機能としてはスペルチェック，べた書きの文書をビジネスレターの形式に変換するオートフォーマット機能，ペン1本で文章の一部入れ換えができるドラッグ&ドロップ機能などが用意されている。

通信機能としては，PCMCIAに準拠したカードスロットを装備し，FAXモデムカードとセルラーフォンを組み合わせてFAX送信や情報アクセスが可能。また，デジタルASK方式高速光通信ポートを装備し，「ZR-5000」同士やIrDAに準拠した光通信ポートを装備したパソコンなどとのデータのやりとりもできる。

ほかにも，スケジュール管理を行うタイムマネージメント機能，さまざまなデータをまとめて整理できるファイラー機能などが用意されている。

「ZR-5000FX」は「ZR-5000」とFAXモデムをセットにしたもの。

価格は「ZR-5000」が749ドル，「ZR-5000FX」が849ドル。

<問い合わせ先>

シャープ㈱　☎06(621)1221, 03(5261)7271

カラーイメージスキャナ GT-8500ART/WIN セイコーエプソン

GT-8500

セイコーエプソンはカラーイメージスキャナ「GT-8500ART」「GT-8500WIN」を発売した。

両機は解像度が400dpiでA4サイズまでの原稿に対応している。階調表現は，読み取り時に1色につき1024段階の階調から最適な256階調を選択して表現する。読み取り速度については従来機「GT-8000」に比べ約1.5倍(同社調べ)となっている。また，最大30枚までの原稿を自動給紙できるオートドキュメントフィーダ，最大A4サイズまでのポジフィルムなどが読み取れる透過原稿ユニットなどがオプションで用意されている。

「GT-8500ART」には「Adobe Photoshop 2.5.1L.E.」(限定英語版)が「GT-8500WIN」にはMICROGRAFX「PICTURE PUBLISHER3.1L.E.」(限定英語版)が標準添付されている。

価格はどちらも128,000円(税別)。

<問い合わせ先>

エプソンインフォメーションセンター

☎0424(99)7133, 06(399)1115

FAXモデム MC14400FXe マイクロ総合研究所

MC14400FXe

マイクロ総合研究所はFAXモデム「MC14400FXe」を発売した。

同機はデータ通信系として最高通信速度が14,400bps，データ圧縮機能はMNPクラス5，ITU-T V.42bisに準拠し，エラー訂正機能はITU-T V.42bisに準拠している。FAX通信系としては最高通信速度が14,400bps, Class 1/2でG3に対応している。設置方法は縦置きと横置きが選べる。

価格は19,800円(税別)。

<問い合わせ先>

㈱マイクロ総合研究所　☎03(3274)2731

パーソナルワープロ WD-Y390 シャープ

シャープはパーソナルワープロ“書院”「WD-Y390」を発売した。

同機は作りたい英文の分類と文書の種類を選び，画面に表示される日本語の質問に答えていくだけで英文レターを作成できる自動英文作成機能を搭載している。欧文モードでは，書体を10種類内蔵し，単語単位

WD-Y390

で自動改行するワードラップ処理，単語間をドット単位で調整して行末を揃えて印刷する両端揃え印刷などが行える。また，正しいスペルを判断する189,000語の英文スペルコレクターも搭載された。

日本語モードでは，新スーパーアウトラインフォント11書体，イラスト入りの文書を作成するためのアプリケーション，各種印刷機能，表計算機能，タイピング練習機能などが用意されている。ほかにも，辞書機能として国語辞典約42,000語，英和辞典約62,000語，和英辞典約41,000語が収録されている。

価格は145,000円(税別)。
＜問い合わせ先＞
シャープ㈱　☎06(621)1221,03(5261)7271

デジタルスチルカメラ
VC-1000Ⅱ/VC-1100Ⅱ
オリンパス光学工業

VC-1100Ⅱ
オリンパス光学工業はデジタルスチルカメラ「VC-1000II」「VC-1100II」を発売した。

「VC-1000II」は記録/再生一体型のデジタルスチルカメラである。記録はJEIDA規格に対応したPCカード(2Mバイト)を使って行い，標準圧縮で31コマの保存が可能。圧縮伸長方式にはJPEG，ファイル形式はDOSフォーマットを採用している。再生画像は本体の液晶ファインダで確認できるほか，PCカードのデータをパソコンの画面などで確認できる。撮影に関してはフルオート撮影，2倍ズーム，マクロ撮影などの機能を搭載している。

「VC-1100II」は「VC-1000II」の機能に加え，本体に画像を送受信できる機能が組み込まれており，外部モデムを介して一般公衆回線もしくは携帯電話(アナログ,デジタルともに可)に接続するだけで画像伝送が行える。伝送速度は一般公衆回線で9,600bps，携帯電話で2,400bpsが見込めるが，回線状態によって異なる。

価格は「VC-1000II」が198,000円，「VC-1100II」が248,000円(ともに税別)。
＜問い合わせ先＞
オリンパス光学工業㈱　☎03(3251)8028

ラベル印刷機
KL-8000
カシオ計算機

KL-8000

カシオ計算機はラベル印刷機「KL-8000」を発売する。

同機は解像度が400dpiのプリンタを内蔵している。印字用のフォントに48ドット文字を採用し，書体も4種類(平成ゴシック体，平成明朝体，毛筆体，丸ゴシック体)から選べる。ほかにも，印刷文字を淡く仕上げる薄文字印刷やバーコード印刷，拡大印刷などさまざまな機能が用意されている。また，6桁×4行の液晶画面を搭載することで，書体や文字装飾などの印刷イメージを確認することができる。

価格は34,800円(税別)。
＜問い合わせ先＞
カシオ計算機㈱　☎03(3347)4811

INFORMATION

第1回
助成公募のお知らせ
ローランド芸術文化振興財団

ローランド芸術文化振興財団は「電子技術を応用した芸術文化」にかかわる優れた活動に対しての助成公募を行う。

今回の助成対象は「音楽」分野に関する活動。例としては，電子楽器を使用したコンサートなどの公演，電子楽器を活用した音楽の普及を目的とした講演会など，電子技術を活用した音楽表現法や音楽教育法に関する調査，研究。対象期間は1995年7月から1996年6月の間に実施されるもの。

募集日程は1995年4月14日までで，結果発表は6月下旬。助成金額は1件あたり10万～50万円。
＜問い合わせ先＞
㈶ローランド芸術文化振興財団事務局
☎06(345)7340

コンピューターゲームの見本市
Game Expo'95
Game Expo'95事務局

3月24日～26日の3日間にわたって「Game Expo'95」が開催される。

同イベントの展示会場ではコンシューマ機やパソコンなどのハードウェアはもとより，新作，旧作，発売予定のソフトを体験できる。また，カンファレンスではゲームデザイナーなどのプレゼンテーションが行われる予定。

会場は展示会が幕張メッセ展示ホール，カンファレンスほかは国際会議場。入場料は前売りで展示会が800円，カンファレンスが4,000円。
＜問い合わせ先＞
日本工業新聞社営業局内
Game Expo'95事務局　☎03(3273)6144

リニューアルオープン記念
バックナンバーフェア
三省堂書店

三省堂書店神田本店では5階のコンピュータ書売り場のリニューアルオープンを記念して，バックナンバーフェアを3月31日まで開催する。

同フェアでは60種類のコンピュータ雑誌を1年分約6,000冊のバックナンバーを一堂に集める。
＜問い合わせ先＞
三省堂書店神田本店5階　☎03(3233)3312

FILES Oh!X

このインデックスは，タイトル，注記──著者名，誌名，月号，ページで構成されています。季節の変わり目です。なにやら今年はいろいろと騒がしいようです。健康はもちろんのこと，身の安全にも気をつけてくださいね。

参考文献
I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
電撃王　主婦の友社
PIXEL　図形処理情報センター
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶NEWS&VIEWS
インテルのPentiumのバグ騒動を追う。バグの内容とメーカーの対応をレポートし，各メーカーの思惑を探る。──編集部，ASAHIパソコン，2・1号，10-11pp.

▶NEW PRODUCTS
115,200bpsの転送速度を可能にしたサン電子のモデム「MS288AF」ほか，ハードとソフトの新製品情報。──編集部，ASAHIパソコン，2・1号，12-13pp.

▶バージョン6の対決！
Windows用の3大ワープロソフトを徹底比較。「一太郎」「Word」「WordPerfect」を取り上げる。──福井香，江畠俊彦，ASAHIパソコン，2・1号，14-27pp.

▶98ユーザーのためのマッキントッシュ教室
PC-9801とMacintoshのアーキテクチャの違いをパソコンの歴史を振り返りながら探っていく。──荻窪圭，ASAHIパソコン，2・1号，28-31pp.

▶「数字略語」大全集
パソコンに関する数字がらみの略語の読み方とその意味を紹介する。──鍛治信太郎，ASAHIパソコン，2・1号，130-138pp.

▶GroballInterface News
「インターネットで教育は変わるの？」と題して，いま肯定的に取り上げられることの多いインターネットを使った教育の実態を紹介する。──室謙二，ASAHIパソコン，2・1号，148-149pp.

▶HEAD LINE NEWS
「ポピュラス」のピーター・モリニュー氏へのインタビューほか，通信ソフト内蔵ザウルスなどハード新製品やイベント情報など。──編集部，コンプティーク，2月号，24-29pp.

▶パソコンゲーム復権宣言
パソコンゲームのメリットについてクリエイターへのインタビューなどを通じてもう一度考える。──編集部，コンプティーク，2月号，35-45pp.

▶巷のメディアミックス
滋賀県立近代美術館で「ファンタジーＲＰＧイラストレーション展」が開催された。作品を紹介しながら，美術とゲームイラストの接点を考える。──編集部，コンプティーク，2月号，128-131pp.

▶NEWS COLLECTORS
アップルのPippin規格発表，SEGA SATURN内蔵のカラオケシステム登場などのニュースとSCEの佐伯雅司氏へのインタビュー。──編集部，電撃王，2月号，18-23pp.

▶とことん楽しむプレイステーション＆セガサターン
発売初日の販売データから両陣営の滑り出しを推測。「AⅣ」「MYST」などの注目ソフト大研究やアクティブスピーカーを使ったシステムの提案など。──編集部，電撃王，2月号，24-45pp.

▶これでいいのか!? バーチャルボーイ
任天堂の「バーチャルボーイ」を発売前に検証する。──編集部，電撃王，2月号，46-49pp.

▶Flash NEWS
リコーのカラーイメージスキャナ「CS-600」などハード，ソフトの新製品情報。──編集部，マイコンBASIC Magazine，2月号，46-49pp.

▶特集　namco BEST HIT GAMES
ナムコの最近のアーケードゲーム「ガンバレット」「サイバーコマンド」などを紹介する。──編集部，マイコンBASIC Magazine，2月号，135-141pp.

▶Arcade Game Graffiti　第12回
1982年に市場に出たアーケードゲームを振り返る。「なめんなよ」「フライボーイ」などが登場。──編集部，マイコンBASIC Magazine，2月号，152-155pp.

▶「P5」互換チップの「P6」機能先取り
インテルの次期プロセッサ，サイリックスやAMDのPentium互換プロセッサをとりあげその構造や性能を明らかにする。──田嶋孝行，I/O，2月号，44-47pp.

▶バーチャル・リアリティとは
バーチャルリアリティの定義から必要と思われる機器の紹介，さらにどんな分野にどのような影響をもたらすかを考える。──ことえり，I/O，2月号，49-51pp.

▶Virtus VR
「Virtus VR」というソフトを使いながら，バーチャルリアリティについて考える。──矢野光一，I/O，2月号，52-54pp.

▶ビギナーのための「映像」&「画像」取り込み入門
画像取り込み機器としてスキャナ，デジタルスチルカメラ，キャプチャボードなどを取り上げ，それらを利用した画像取り込みの実際を解説する。──編集部，I/O，2月号，55-70pp.

▶MultiMedia Watching　14
今回はインターネットを取り巻く日本の事情を解説する。──奥野雅之，I/O，2月号，71-73pp.

▶'94年のパソコン界を概括する
マルチメディア，DOS/Vなど94年のキーワードを中心に1994年にパソコンがどう進化したのかを考える。──下田洋，I/O，2月号，98-100pp.

▶コンピューターによるTV会議システム最前線
テレビ会議システムの概要から，それに関わる技術などの最新事情を詳しくレポートする。──嶋是一，I/O，2月号，101-103pp.

▶NEWS BYTES
Pentiumのバグに関する情報やアップルがマルチメディアプレーヤー規格「Pippin」を発表した話題など。──編集部，I/O，2月号，158-161pp.

▶ASCII EXPRESS
「Windows Solutions Tokyo 94」のレポートや各種新製品情報，インタビューなどを満載。──編集部，ASCII，2月号，209-230pp.

▶The Ranking Cruise
11月期のハード，ソフト，パソコンに関する書籍などのランキングが一目でわかる。──編集部，ASCII，2月号，231-240pp.

▶特集　見えてきた最新OSと次世代プロセッサ
最新のOSと次世代と注目されるCPUをいくつか取り上げ解説する。──編集部，ASCII，2月号，241-264pp.

▶特集2　Fun Computing
Windows，PC-9801上で動作するゲームの紹介やそれらに関係したハードの話題など，ホビー関連の情報が満載。──編集部，ASCII，2月号，281-306pp.

▶魅惑のニューテクノロジー　第11回
「ソフトウェアMIDI」と題して，ソフトウェアによる音楽演奏の可能性について実例と理論を織り交ぜ解説していく。──編集部，ASCII，2月号，346-351pp.

▶インターネット膝栗毛　ROUTE 1
最新トレンドなどインターネットに関することを紹介する。──編集部，ASCII，2月号，356-359pp.

▶スペシャルインタビュー22
電卓の元祖「14-A」を生み出した4兄弟のひとり，カシオ科学振興財団理事長の樫尾幸雄氏に話を聞く。──遠藤諭，ASCII，2月号，385-390pp.

▶稀代もののけ考
栓抜きやクリップからコンパクトカメラまで巷で見つけた優れモノを紹介する。──バカパパ，ASCII，2月号，412-413pp.

▶なんでも相談室
カラー液晶ディスプレイの違いや電話回線についての質問などに答える。──内田顕弘ほか，ASCII，2月号，424-427pp.

▶ワンステップ通信
障害者が自立するための道具として使うコンピュータを考える。──編集部，ASCII，2月号，438-439pp.

▶特集　MO「光磁気ディスク」の導入と上手な活用
MOの導入と活用を初歩から各機種別の特徴にいたるまで丁寧に解説。──編集部，My Computer Magazine，2月号，8-26pp.

▶レッツプログラム
ひとつのテーマを決め読者からプログラムを募集し，そのプログラムを解説する。今回は「ポイントシールの組み合わせ」。──藤本健，My Computer Magazine，2月号，85-92pp.

▶初心者のためのパソコン研究室　2
よいディスプレイの条件を考えたり，ディスプレイ選びのポイントを伝授したりする。──SpaceClub，My Computer Magazine，2月号，101-103pp.

▶CD-ROM NEW STYLE
CGクリエイター川口洋一郎氏の作品を網羅したCD-ROM「COACERVATERA」などを紹介する。──編集部，LOGIN，3号，38-43pp.
▶ハードウェアFLASH！
アイワの「PC-MT466V2」，エプソンのプリンタ「MJ-5000C」などの新製品を紹介する。──編集部，LOGIN，3号，44-47pp.
▶THE NEWS FILE
「マルチメディア'94」のレポートやアップルのPippinの紹介など。──編集部，LOGIN，3号，48-53pp.
▶史上最強のCOMDEX FALL '94レポート
11月14～18日に開催された「COMDEX FALL '94」のレポート。──編集部，LOGIN，3号，194-197pp.
▶インターネットを遊べ!!
話題のインターネットをキーワードから解説したり，アクセス方法を紹介する。──編集部，LOGIN，3号，202-205pp.
▶Microsoftに魂を売れ！
Microsoftの歴史，製品を紹介し，立派な「マイクロソフター」となることを推奨する。──編集部，LOGIN，3号，208-211pp.
▶架想楽園へ行こうver.2.04
松下電工が提唱する21世紀住宅開発プロジェクト「バーチャルハウス」を紹介する。──中田宏之，LOGIN，3号，212-215pp.
▶くねくね科学探検隊　第13回
「囚人のジレンマ」というパラドックスを取り上げ，「裏切り」のもつ意味を東京大学教養部助教授池上高志氏に聞く。──鹿野司，LOGIN，3号，220-223pp.
▶特集　マルチメディアソフト制作の現場から
マルチメディアソフトを制作している会社の現状を紹介する。「マルチメディア'94」のレポートもある。──編集部，PIXEL，2月号，93-114pp.

MZシリーズ

MZ-2500(BASIC-M25)
▶ドット落ち
いままでとはひと味違う落ちものゲーム。──Sweet Mint，マイコンBASIC Magazine，2月号，81p.

X1/turbo/Z

X1シリーズ
▶ドッグ ファイト
対戦型シューティングゲーム。──木村和明，マイコンBASIC Magazine，2月号，99-100pp.
▶ぷよぷよ
対戦のときの音楽プログラム。NEW FM音源ドライバ用。──川村賢治，マイコンBASIC Magazine，2月号，112-113pp.

X68000

▶Super Soft Express
話題のソフトを紹介。X68000用は「パックランド」など。──編集部，コンプティーク，2月号，47-69pp.
▶電撃新作予定表
今後発売予定のソフトを，機種別に掲載。X68000用は「プリンセスメーカー」など。──編集部，電撃王，2月号，156-161pp.
▶爆弾野郎
壁を爆破させていくアクションゲーム。──吉村光司，マイコンBASIC Magazine，2月号，101-102pp.
▶HELL
機雷などをよけながら，深海1万メートルまで潜っていくアクションゲーム。──N.Yusa，マイコンBASIC Magazine，2月号，103-104pp.
▶SUPER SOFT Hot Information
新作ソフトの紹介。X68000用は「パックランド」のニュースなど。──編集部，マイコンBASIC Magazine，2月号，とじ込み付録10p.
▶なんでもQ&A
Human68kのリダイレクト機能を使用したOPMファイルの演奏についての質問に答える。──シャープ，My Computer Magazine，2月号，140-141pp.
▶AV STRASSE
計測技研から発売された「シャーペンワープロパック」の特徴をわかりやすく紹介する。──編集部，ASCII，2月号，367-368pp.
▶ONLINE SOFTWARE INDEX
大手ネットにアップロードされたソフトを紹介する。X68000用は2HDディスクの内容をそっくりそのままひとつのファイル(ディスクイメージ)にする「dpack.x」など。──編集部，ASCII，2月号，451p.
▶SX-WINDOWプログラミング　第16回
今回はローカルリソースに対応した開発環境を構築するためにヘッダファイルを修正する。──吉野智興，C MAGAZINE，2月号，126-131pp.

ポケコン

PC-E500
▶ドットット
壁にぶつかると跳ね返るドットを操作してゴールを目指すワンキーゲーム。全8面。──マイコンBASIC Magazine，A.SIR，2月号，105-106pp.

新刊書案内

電脳曼陀羅
中村正三郎著
ビレッジセンター刊
☎03(3221)3520
四六判　267ページ
1,200円（税込）

中村正三郎が「電脳騒乱節」に続いて『ざべ』で連載した「電脳曼陀羅」は諸般の事情でいきなり中止になったわけだが，その「電脳曼陀羅」とその連載中断に伴うドキュメントを収録したのが本書だ。多くの読者は「電脳曼陀羅」よりもその連載中断がらみのドキュメントを読みたがるだろうし，本書も「電脳曼陀羅ドキュメント」としたほうがしっくりくるくらいだ。実のところ，前半の取材に基づくウゴウゴルーガの話やJRAの話より，後半のマイクロソフトがらみの話から連載中断に至るまで，つまり，著者のプレイフィールド内の話のほうがずっと面白い。ある程度内部事情を知った上で思い切ったことを書くからだ。逆に，思い切ったことを書くにはある程度インサイドな雰囲気を感じさせる必要があり，回によって面白さに差が出る原因はそのあたりにある。
著者の人気の秘密は知識の広さを文章と発想のうまさでパソコン界に結びつける強引さにあるわけだが，それ以上にくすぐり方の巧みさが凄い。ある程度パソコンを使い込み，業界動向に詳しい人をうまくくすぐるのだ。知っている人にしかわからないギャグの創造がうまいのである。そして多くの人が書けない，あるいは気づかない視点で思い切ったことを書く。極端な書き方をして書かれた人の気分を害することがあっても，概してパソコンマニアは極端な言い回しを好むため，そのほうが受けるのだ。
それにしても，1冊にまとめられたものを読むと，どんな裏があったにしろひどい話であったことがわかる。と同時に事件の不可解さは消えない。不可解なまま終わったのだ。著者もそのもやもやが全然晴れていないのだなぁと，その後のメッセージを見て思う。ただ，極めて日本的な事件だったのを，著者が日本的な処理をこばみ，パソコン通信を使って表沙汰にしたのは非常に特筆すべき事柄である。　(K)

世界のコンピュータマップ'95
ジャストシステム出版編集部編
ジャストシステム刊
A5判　281ページ
2,000円（税込）

コンピュータをめぐる技術の進歩はあいかわらず速いが，そのことが重要なわけではない。使う側が満足のいくものであるかが重要なのだ。
本書はコンピュータが世界の中でどのような状況にあるのかいろんな視点で紹介する。主な内容はアメリカやヨーロッパ，アジアのコンピュータ業界の現状，日本とアメリカのパソコンをめぐる比較，各都市でのコンピュータの活用例など。
ここで書かれている状況も1年後には変わって新たな流れが生まれているかもしれないが，いまコンピュータを利用して世界がどんな状況にあるのか知っておくのもいいだろう。

「超」小さい物質の秘密
山吉恵子著
柳田博明監修
オーム社刊
☎03(3233)0641
B6判　148ページ
1,500円（税込）

日常生活でよく聞く言葉だからといって必ずしもそれについて知っているわけではない。「超微粒子」という言葉聞いたことはないだろうか？　フロッピーディスクや化粧品などにも使われている。言葉のとおり「とても小さい粒子」というくらいの想像はつくのだが。
本書はその「超微粒子」について，2人の登場人物による対話形式で専門用語をなるべく使わず，わかりやすくその世界へと案内してくれる。「超微粒子」の定義から性質，生成法などについて触れている。こういった分野に興味のない人にこそ一読してほしい。

QUESTION and ANSWER

シャーペンを使用しているのですが，シャーペンのウィンドウをひとつも開いてない状態で起動するとハードディスクをアクセスした状態が10～15秒ほど続いてしまいます。ファイルをひとつでも開いていれば，次のファイルの起動は瞬時に行われます。おそらくハードディスクをCドライブに設定していることが原因だと思うのですが，どこを直せばいいのかわかりません。SXWIN.ENV，STARTUP.ENVなどのパスは書き換えましたし，一応マニュアルはひととおり目を通したつもりですが，シャーペンに環境変数のようなものはなさそうなので，SX-WINDOWのパスをたどっていると思うのですが。　神奈川県　須藤　泰賢

編集部でも須藤さんと同じ設定を行って試してみましたが，それほど立ち上げ時間はかかりませんでした。試しにSXWIN.ENVのパスを全然違うところに切ってみても大差はありませんでした。

SX-WINDOWはファイルが見つからない場合はAドライブから順にサーチしていきますので，後ろのほうのドライブや込み入ったディレクトリ構成のドライブだと検索にかなり時間がかかることがあります。

となると起動に必要なファイルがシャーペンと同じディレクトリにないのではないかという疑いが出てきます。一応，シャーペン.ARCが同じディレクトリにあることを確認してください。また，関係ないとは思いますが，環境アイコンのメニューの下にある「外部コマンド設定パス」も確認してみてください。

それでもどうしようもない場合は，起動パスを特定してやるのが手っとり早いかもしれません。デスクトップメニューの「文書編集」やプルダウンメニューの「内容表示」，各文書ファイルの起動コマンド名を絶対パスつきで指定してみてください。また，こういった指定でパスが違っていると起動が遅くなりますので，そのへんにも注意してください。

ちなみに，シャーペン.Xはシャープ純正のアプリケーションにしては珍しくちゃんとリエントラントな構成になっていますので，現在すでにシャーペンが開いている場合にはメモリ上にあるシャーペンのプログラムがそのまま呼び出されるのです。とすれば2度目以降の起動が速いのは当然の話ですね。

1月号のローテク工作実験室「X680x0周りのSCSIを探る」を読んでSCSIについてよくわからなくなりました。私はてっきりX68030のSCSIはSCSI2だとばかり思っていました。それなら1993年9月号のMO特集に載っていたようにSCSI2対応のMO機器がたいていX68030につながるというのはなぜでしょう。本体がSCSI1ならばSCSI2からINQコードが決められたMO機器は簡単にはつながらないはずだと思うのですが。

それから，ハードディスク上にA001からA100までのファイル（たとえばDōGAのPICファイル）が順不同で記録されているとします。これらのファイルを番号順にソートしてほかのハードディスクへコピーするにはどうすればよいのでしょうか。つまり，ディスク上に番号順に記録されているようにしたいのです。Human68kのコマンドにはそれらしい機能を持ったものはないようですが。

3番目の質問です。先日，中古のMOドライブを格安で入手しました。エレコムのEMO-128という機種ですが，正面のパワーランプなどのようすは1993年3月号に載っていたGMD-128という機種とまったく同じものですのでおそらくドライブも同じものではないかと思いますが，マニュアルがついていなかったので詳しい設定がわかりません。MOのディップスイッチはSCSI-IDを2，Modeを7，TerminatorをON，ParityはOFFで使っています。いまのところちゃんと読み書きはできているようですが，設定はこれでいいのでしょうか。SX-WINDOWでディスクをイジェクトして別のディスクを入れるとどのアイコンもマウスのクリックに反応しなくなってしまいます。また，DōGAのPICファイルをMOにコピーしてアニメーションさせると，うち何枚かは画像が崩れてしまいます。　福岡県　小豆嶋　修

小豆嶋さんからは3種類の質問をいただきました。順にお答えしましょう。

X68000のSCSIはSCSI1をベースにMOドライブを拡張したものと考えればいいでしょう。SCSIボードの発表が純正MOの発表と同時だったことを考えれば当然のことといえるでしょう。SCSI2ではちゃんとMOドライブがサポートされていますが，SCSI2規格ができる前からMOドライブは存在していたわけで，考えてみれば光磁気ディスクとはいっても，OSにとってみれば扱いはリムーバブルハードディスクと変わるものではありません。SCSI2でINQコードは新設されましたが，SCSI規格の範囲内でアクセスできないものではなかったのです。実際，初期のMOドライブはハードディスクとして使用するためのINQコードを出力するようになっていました。それらは当然SCSI1で接続されていたわけです。X68000のSCSIシステムではMOをいったんハードディスクとして認識しておき，SCSIドライバが装置を調べてMOかどうかを決めるようになっていました。

INQコードというのは機器を自動識別するための手段であって，手動で設定してやればそれ用のドライバを使って処理できなくはありません。

普通ならINQコード不明でも，あとはデバイスドライバでのフォローを期待するような構成になっているので新しい機器でもなんとか接続できるようになっています。多くのパソコンでMOの接続キットにわざわざドライバが付属しているのはそういうわけです。

X68000の場合，そういった特殊なドライバを必要としなくてもMOの使用ができるようになっているのですが，最近のMOがつながらないのには少し特殊な事情があります。

SCSIボードをつけるとSCSI IOCSが拡張されます。SCSIドライバはそのコールを使って作られています。その追加IOCSはどこからきたかというと，SCSIボード上のROMに入っているわけです。

SCSI規格が拡張されてもSCSIドライバが単にSCSIボードのROM（古い規格で作られている）上のルーチンを呼び出しているだけなので，新しい機器への対応ができないのです。X68030以降ではSCSI2のMO用INQコードが追加されIOCS ROMが改善されましたが，従来の機種用のSCSI ROM部分はバージョンアップされていませんので新しいINQコードしか持たない製品はSCSI IOCSの段階ではじかれてしまうのです。

これを解決するには，SCSI ROMをX

68030仕様のものに交換するか，SCSIドライバを新しくして，SCSI ROMに依存しなくなるようにしなければなりません。

ROM交換は大変ですのでHuman68k ver.3.0の発表とともにSCSIドライバ（またはHuman68k本体）が拡張されるべきだったのですが，それが行われなかったのです。同時に発表されたX68030のROM部分はちゃんと拡張されていますのでおかしな話ではあります。

SCSIドライバに依存しないSxSIドライバやSCSIドライバをごまかすINQPATCH.Xは新しいMOのINQコードに対応したものですので，それらを使えばどんなMOでもアクセスできるわけです。

ということで，X68030用のSCSIはSCSI2ではありません。SCSI2というのはMOやCD-ROMが拡張された以外にもたくさんの改良や拡張が行われています。X68000のSCSIはSCSI1にMOが接続できるようにしたもの，X68030のSCSIはそれをさらに改良したものというのが正解でしょう。

では続いて2番目の質問です。

フリーソフトウェアではディレクトリソートツールなど，そのような目的のためのツールもありますが，私ならたぶんバッチファイルを作るか，SX-WINDOWを立ち上げてディレクトリウィンドウを名前順でソートしてからコピーします。

小豆嶋さんのおっしゃるように標準コマンド一発では対処できない問題なので，多少の手間はかかります。具体的な操作を逐一挙げてみましょう。

COMMAND.Xは/Nオプションで名前順にディレクトリ表示を行いますので，それをリダイレクトして加工してやればあとは置換とキーボードマクロをちょこちょこ使うだけでなんとかなります。ここではDōGA PICファイルをCドライブのカレントディレクトリに転送する場合を考えます。

```
A>DIR /N *.PIC  >TEST.BAT
```

でディレクトリ内容をファイルに落とし，

```
A>ED TEST.BAT
```

でエディタに読み込みます。ちなみに同様に，/Tオプションで日付順にソートして出力です。

先頭の不要行をカットし，ファイル長表示などの不要部分を消します。カーソルを最初の行の拡張子の後ろにあわせ，

```
ESC @         （マクロ開始）
CTRL-K        （カーソル以降削除）
↓             （カーソル下）
UNDO          （マクロ実行）
```

で最終行まで処理が終わったらESCを押して処理を止めます。

まず，

```
" PIC"        （スペース＋PIC）
```

を，

```
".PIC"
```

に置換し，さらに，

```
" "           （スペース）
```

を，

```
""            （すぐリターン）
```

に置換します。これでファイル名の並びが作られたはずです。

最初のファイル名先頭にカーソルをあわせて，

```
ESC @
"COPY "       （文字列を打つ）
CTRL-P        （カーソルを最後に）
" C:"         （ドライブ名を指定）
→             （カーソル次の行に）
UNDO
```

で文字列を加工して，最終行まで終わったらESCキーで処理を止めて，ゴミを削ります。

加工の方法にはほかにもいくつかスタイルがあります。キーボードマクロを使いこなすようになればさらに複雑な処理も簡単にこなせるようになりますので，各自で工夫してみてください。行の二重化や文字検索を絡めるとたいていのことはできます。

それからSX-WINDOWを使うほうはなんの説明もいらないでしょう（シフト＋メニューはみんな知ってますよね？）。

3番目の質問にいきましょう。

エレコムのMOドライブはオリンパスのOEM品と思われますので製品としてはまったく同じものと考えてもいいでしょう。

見たところ特にまずい設定をしているわけでもなく（むしろきわめて適切な設定です），ちゃんとアクセスできているのならそれでかまわないと思います。確か，マニュアルにある標準的な設定を行うとX68000では使用できなかったはずですので。

MO側のファームウェアの問題とSX-WINDOW側が想定しているドライブの問題があります。SX-WINDOWではメディアの存在を確認するため，頻繁にチェックを行っているわけですが，ドライブが使用できるかというチェックをしたときの応答がきわめて遅かったり，実際にはREADY状態でないのにREADYを返してきたりする製品が存在するのです（MOを使うたいていの局面では特に問題がないのですが）。

MO特集で性能的にかなり劣っていてもソニードライブを推奨していたり，価格的に飛び抜けていても東芝ドライブを推奨していたのにはそれなりの理由があります。ファームウェアはどんどんバージョンアップされているので，できるだけ新しい製品を買ったほうがよい，というのも書いていたと思います。

さて，こういったMOをSX-WINDOWで使用するためにいくつかの技が開発されています。すなわち，イジェクトボタンをクリックしたあとすぐにメニューやダイアログを開いて，メディアの交換が完全に終了するまでSX-WINDOWの流れを停止させるのです。ドライブをクローズしたあとなら，なにかダイアログを開いている間に交換してしまう手もあります。

DōGAのPICファイルが崩れるというのは読み込みエラーが発生していると思われますので，ParityをONにする，ヘッドクリーニングする，SCSI機器の接続順を変える，SCSIケーブルを揃える，ターミネータ(外部)を確認するなどの対処をしてください。いつも同じ画像が崩れるならメディアを疑ったほうがよいでしょう。（中野修一）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町 3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

暖かいと思ったら急激に寒くなったり，この時期はとかく天気が不安定です。それでも，暖かい日にはこたつむりから脱皮して元気いっぱい遊びましょう。新しい進路について悩んでいる人も，たまには息抜きしないとまいっちゃうからね。

◆1月号のCD-ROM特集はとてもよかったです。私はドライバすらもっていないのにCD-ROMを買ってきて，デバッガでCD-ROMのファイルインデックスを読み，画像ファイルを読み込んでいた時期がありました。いまは，CD-ROMドライブをつなぐだけならなんの問題もないのですから，もっとX68000ユーザーに普及してほしいものです。そして，X68000ユーザーのためにOh!X集大成CD-ROMを付録にほしいですね。
竹本　隆博(32)京都府

◆う〜ん，予想どおりのCD-ROM特集といったところですかね。結局，絵や音のデータをどれだけ生かせるか，というところでしょうか。個人的には他機種のデータフォーマットを勉強できた点がよかったですね。島崎 智博(31)福岡県

X68000では，割り切って使うことで利用価値が見えてくるCD-ROMですが，今度は，読者の皆さん自身で利用価値を探ってみませんか？

◆CD-ROMドライブを購入しました。「ゆみみみっくす」をX68000でプレイするためだけに……。私はメガドライブはもっていますが，メガCDはもっていないのです。しかし，メガCD用の「ゆみみみっくす」はもっていました。そして，今回のCDS-Eの購入……メガCDを買ったほうがずっと安くついたような。う〜む。
太田　崇貴(23)岡山県

太田さんは，X68000で再生することに魅力を感じたのですから選択は間違っていないと思いますよ。それに「ゆみみ」以外にもCD-ROMの使い道はいろいろあるでしょうから，高い買い物ではないはずです。

◆1月号の「割り切って使うCD-ROM」の特集を読み進めるにつれ，つくづくいまのX68000が中途半端な存在になりつつあると感じました。画像表示能力，音源，スピードなど，いろいろな面で転換期にきているのかもしれません。それでもX68000が好きなのですが。
深見　満彦(17)和歌山県

このようなユーザーの期待を一身に受けるメーカーがどう動くか。シャープさんがんばってくださいよ。

◆「付録CD-ROM」より「付録MOディスク」のほうがいいと思う。　　一ノ瀬　宣彦(23)群馬県

そうなると値段（コスト）もさることながら，雑誌にMOを付録としてつけていいのだろうか，という疑問も生まれます。確かに普及率からいえば断然MOのほうが有利ですけど。

◆1月号の「ローテク工作実験室」はよかった。記事中で「ほかにこんなことをする人がいるのだろうか」なんて書いてありましたが，私もやろうとしているひとりです。某アニメのためだけにCZ-6VS1とQUANTUMの1Gバイトハードディスクを買うつもりでいたのですから（で，PCMのためだけにAWESOME-Xの購入も検討中）。しかし，QUANTUMドライブが接続できないのには困ったものです。ハードディスクは500Mバイトにして，他メーカー品を探さなければならないのかな（でもQUANTUMドライブをつなげたいよ〜）。まあなんにせよ，今回の記事には助かりました。　　若月　功(20)埼玉県

若月さんの実体験レポートも期待してます。

◆2月号の特別企画で次世代ゲーム機を扱うみたいですけど，もうX68000では誌面が埋まらないということでしょうか？　だとしたら少し残念です。X68000はフリーソフトウェアがたくさんあるわけですし，そういったコーナーもやってほしいものです。立花　大助(19)奈良県

大丈夫。今月はいつものとおりのOh!Xに戻ってます。

◆最近仕事の都合もあってPC-98やDOS/Vマシンを使うことが多くなりました。一応，世間の流れも知っておくべきだと思い，WINDOWSプログラミングの勉強も始めてしまいました。そこで思ったことは，WINDOWS関連の資料は本屋にたくさん並んでいますが，そのほとんどがビギナー向けでプログラマ用の本がほとんどありません。事実，CGデータのBMP形式について知りたいと思っていたら，その情報をOh!Xで読むとは……。これからもいろいろな情報源として愛読していくつもりです。米原　孝太(24)神奈川県

最近では，パソコンを使う＝ソフトを使う，ということですから，システムの内部情報まで知ることは難しいかもしれませんね。読者の皆さんも，なにか知りたいことがあれば気軽にアンケートハガキに書いてくださいね。

◆SX-WINDOW付属のシャーペンでレポートを書いていますが，コツがわかるとなかなか使えますね。数式も外字を使ったり，強制改行幅と文字倍率で分数やルートの混在したものも書けてしまいます。あとほしいのはグラフ描画機能ですね。これさえあればいまのところなんの問題もないのですが。「DatacalcSX-68K」に期待，といったところでしょうか。数カ月かけてのレポートをしてもらいたいものです。「Super BUSINESS」も発売されるようだけど，どんなものか気になるし。　　田辺　和也(20)神奈川県

グラフ機能はありませんが，計測技研から発売された「シャーペンワープロパック」は，なかなかよさそうですよ。田辺さんは購入されました？

◆「パックランド」を買いました。とてもよくできていますね。実際，アーケード版との比較は年月がたちすぎてわかりませんが，「よくできている」って感じがします。ムチャクチャ新

◀岡村　直也　兵庫県
意外な適性を見せる恵子ちゃんを眺めている場合ではないぞ、学君！　ぼやぼやしていると恵子ちゃんの尻に敷かれてしまうぞ（なんだそりゃ）。

◀岡村　直也　兵庫県
本当に無事でよかった。それにしてもあの兵庫県南部地震には、関東にいながらも心底ビビりました。被災者の皆さん、がんばってくださいね。

作ソフトが減っていますが，こういう感動できるソフトが出るかぎり，X68000を使っていて十分に満足できます。　多田　哲也(23)兵庫県

特に電波新聞社のアンソロジーシリーズは，昔からゲームをやってきている人間にとって思い出深い作品ばかり。これからも期待したいですね。

◆最近，MOを接続しました。とはいっても買ってきたのではなく，会社からの借りものですけどね（休日だけは俺のもの）。さて，接続はしたけど認識させるのにひと苦労。一難去ってまた一難。動作が安定せず，たまにハングまでする始末。なんとか，原因がディスクキャッシュとの相性であることを突き止め，MOが動く環境が整いました。それにしても，1993年9月号と1994年6月号で，MOやX68000の周辺機器のことを詳しく解説してあったOh!Xには，たいへん助けられました。　横田　浩(24)東京都

ちょっと照れちゃうけど嬉しいハガキ。これからも，読者の皆さんに役立つ情報を掲載すべくがんばるぞ。

◆東京システムリサーチから発売された「Xellent30」はたいへん興味深かったのですが，EC030専用ということは，もしかしたら「NetBSD」が動かないということなんでしょうか。だとしたら悲しすぎる〜。ところで期待していた「魔法大作戦」の記事がボロボロだったのでショックを受けてしまいました。今月は「Xellent30」を使った場合の記事なんかを載せてくれるとありがたいのですが。そうそう，68000モード時のコプロはI/O接続になるのですか。

白井　保弘(26)三重県

残念ながら「Net BSD」は動きません。次に「Xellent30」を使っても「魔法大作戦」では，強制的にキャッシュをOFFにしているので逆に重くなります。あと「Xellent30」上の68881は，コプロ接続なので68000からアクセスすることはできません。　(瀧)

◆待ちに待った「魔法大作戦」を買いました。ハードディスクへインストールしている間に説明書に目を通すと「ゲームは1人用です」の言葉が……。少し胸の痛い思いをしながらプレイしてみました。いきなり背景スクロールカクカク，キャラクターもカクカク点滅状態。それを見た僕もカクカク状態に……。一気に血の気が引いてしまいました。う，うそやろ，X68000 XVIなのにそんなこと，あっ10MHzになっているのか，と思い本体を見ると無情にも16MHzのランプが光っていました。打開策はあるのでしょうか。　辻野　雅弘(22)滋賀県

やはりX68030を買うしか……しかしX68030でもキャラ消えは起こるんですよね。

◆僕のマシンは室内温度20度くらいで，使用する1時間前に電源を入れないと起動しません。やはり修理に出したほうがいいのかなあ。そうそう，最近Oh!Xのページ数がお父さんの髪の毛のように薄くなっている気がします。

片山　明義(16)奈良県

がんばって，すくすくと育てなくては。

▲武田　正道　兵庫県
かわいい連れがいると楽しさ倍増のスキー。でも，あの人と行ってこの人と帰って来るような娘は，ちょっと嫌だなあ。

▲岩瀬　貴代美　福岡県
退屈なあなたにお勧めなのが，数の子を一粒一粒分離すること。5分もたてば目が血走るし心臓はバクバク。退屈な思いなんてふっ飛びますよ。

◆MJ-700V2Cを買いました。試しに，昔X1turboZで作ったMZ-1P17用のカラーハードコピープログラムを，一部変更して印刷してみました。4096色画像（取り込み）ですが，結構きれいに印刷できます。メモリなどの関係上，4色分解はできませんでしたが，色にじみもありません。インクジェットも使いやすくなったものです。

吉田　秀行(28)岐阜県

頻繁に性能アップしている状況なので，買い時を見極めるのが難しいところですが，1台ほしいと思わせるものがありますよね。い，いかん，こんなことを考えていると足が勝手に秋葉原へ向かってしまう。

◆ちょっとわけあってX68030ではなくX68000 XVIを買いました。なぜ，という気がしますが，やはり自分に合っているのはX68000 XVIあたりぐらいだな，と思ったからです。そんなおり，X68000 XVI用のアクセラレータの話が！　さっそくツ○モ電機の店頭で見てきました(動作試験中でしたので)。X68030には及びませんが，十分です。もちろん，Oh!Xの記事に書かれていたことは問題です。でも，これはX68030を買っても同じだったわけですから……とにかく，これからもX68000初代とX68000 XVIをうまく使っていかなくちゃと思っています。私にとってX68000 XVIでX68ファミリーも4台目です。長く使っていきたいです。はい。

和田　哲也(25)東京都

ここまで，ユーザーに愛されるマシンというのも珍しくなりました。僕も大切に使っていかなくちゃ。

◆マシンがX68030になりました。しかもメモリはフル実装。でも，わざわざ東京まで行って地元メーカーのメモリボードを買うっていうのは……。　中江　克行(21)石川県

なんとなく，無駄ですねえ。工場直営のできたてほやほやメモリボードの街頭販売でもあったらいいのに。

◆修論締め切りまであと1週間。第1稿を見た先生のお言葉を要約すると「書き直せ」。はたして間に合うのか？　頭をよぎる「留年」の2文字。以下，次号。　松永　貴輝(24)大阪府

間に合わなくても間に合わせるために，命を削ってでもがんばるのだ！

◆横島忠夫のぬいぐるみ（UFOキャッチャーのやつ）のバッグに下着が入っていました。1月号の137ページを読んで，美神令子にはかせられるかどうか試してみましたが，サイズが合いませんでした。ならば……おキヌちゃんのハカマは縫われていた。　高倉　英之(21)愛知県

こうして，パンツに魅せられた男が，ひとりUFOキャッチャーを渡り歩くのであろうか（冗談ですって）。

◆結婚して以来，うちのX68000 ACE-HDは奥さんのゲームマシンと化しております。ハマっているのは「テトリス」「パックランド」「究極タイガー」。いずれも僕のレベルを超えております。おかげでジョイスティックが壊れてしまいました。　坂口　郁夫(27)北海道

なんかハガキを読んでいるうちに，必死の形相でジョイスティックを握っている奥さんと，それを横で眺めている坂口さんの絵が浮かんできて，ちょっとだけ笑ってしまいました。

◆仕事が終わって帰宅し，X68030の電源を入れると“かな”キーと“全角”キーが点灯するではありませんか。どうやら犯人は1歳の息子のようです。机の上に置いてあるX68030のパワースイッチにまで手が届くようになったのです。恐ろしや。　後迫　浩一(34)神奈川県

恐ろしがってはいけません。せっかく興味を示し始めたのですから，いまがチャンスです。子供が寝る前に必ず「Inside X68000」を読んで聞かせ，子守歌の代わりにニーモニックを暗唱して，いまから教育するのです（なんの？）。

◆修学旅行で大阪〜青森間を列車で移動する場合，文部省の取り決めで北海道の学校に日本海縦貫線を利用できる優先権があるそうです。船や飛行機も決まりがあるそうで，おかげで僕の場合，東京経由の列車で片道まる2日かけて北海道まで行くことになりました（でも楽しかったなあ）。そのとき北海道とは“おいしいところ”だとよ〜くわかりました。食い倒れよりすごいかも。　村瀬　正美(19)兵庫県

そうそう，北海道はとってもおいしいとこ

おめでとうハガキありがとう！

◀山口　嘉久　神奈川県
手作り度満点、版画の年賀状をありがとう。しかし、最初に見たときは左右反転を間違ってしまったのかと思っちゃいましたよ。

◀板垣　修　千葉県
久しぶりに登場したと思ったら、新年早々飛ばすなあ、板垣さん。ま、とにかく元気そうでなによりです。今年もよろしくね。

◀伊藤　健文　神奈川県
熱い想いが伝わってくるハガキ……なんですが，文字が小さすぎてメーカーの人に伝わるかなあ。ちょっとだけ心配です。

◀後藤　浩一　大阪府
パワーアップもいいけれど、使っていけばマシンといえども老朽化します。マシンの健康状態には気をつけてあげてくださいね。

◀佐藤　貴是　神奈川県
ひとりじゃあまりにも寂しすぎるので、担当からもおめでとう、と祝ってあげましょう。がんばって、ゲームクリエイターを目指してください。

ろです。海産物の新鮮さといったらもう，たまりませんね。なんて，考えるとおなかがすいてしまったので買い物行こうっと。

◆スタッドレスタイヤを買った。FRの車は冬が怖いから。しかし，雪が降らない。全然降らない。タイヤよりメモリを買うべきだったか？

谷川　正洋(24)広島県

（もぐもぐ←なんか食べてる）雪が降らない（んぐんぐ←なんか飲んでる）のなら，雪が降っているところまで（もぐもぐ←また食ってる）遊びにいくのもいいんじゃないでしょうか（ごちそうさま←食べ終わった）。

◆大学に入ってからついに彼女をつくった。つくったのはいいのですが，現在バイトがない身分。そのため，彼女の誕生日のプレゼント購入のため早朝バイトをするはめに。しかし，誕生日当日，思いっきり喜んでくれた彼女の顔を見て疲れも吹き飛びました。ん～彼女とはいいものだ。

津田　将弘(19)大阪府

幸せそうでなによりです。僕だって，いまはおなかがいっぱいでとっても幸せだもんね（ひがむんじゃないって）。

◆もう年末は忙しい！　やっと車の免許が取れました。若草色の……そしていまだに就職活動を続けています。卒業研究もあるし，必須単位も残っているし，「雷電DX」の練習ステージからエキストラステージにいけないし，「ダライアス外伝」は，まだクリアしていないし。やり残しが多いので，今年も忙しくなりそうです。今年こそよい年にしましょう。

日比生　雅治(22)大阪府

しかし，毎年やり残したことがあると，どんどん蓄積されて，やがて1年中やり残したことを消化し続けるようになってしまったらちょっと嫌だなあ。

◆新年明けましておめでとうございます。今年でX68000は8年目を迎え，シャープが公約していた“8年間アーキテクチャの未変更”の期限切れの年でもあり，新製品に期待を寄せております。ところで，愛機X68000初代の労をねぎらうために，アーベル社のウルティマOAクリーナでボディの洗浄をしました。グレーであるため，汚れが著しく目立っていたのですが，洗浄後美しさを取り戻しました。X68000はスタイルがよいのできれいにしておかないとダメですね。今年は，再びX68000に市民権が与えられる年でありたいと願っています。

大久保　敦(21)大阪府

再び市民権を！　なんて言葉を聞くと，なんだかX68000を使っているのは，パソコン界の異端児みたいじゃないですか。ってそうなのかなあ……。

◆早いもので，社会人になってもう年を越えようとしています。年をとるにつれて1年がどんどん短くなるのは，きっといままでの歴史がどんどん長くなっていくからでしょう。ああ，1994年が過ぎていく。　新井　誠治(23)北海道

あまりじじむさいことをいっていると，本当に無益に時間が流れてしまいますよ。気をつけましょうね。

◆「趣味は仕事にしないほうがいい」とか「趣味と実益は両立しない」などということをよく聞きます。しかし，「趣味とは縁もゆかりもない仕事が，めちゃくちゃ忙しくて趣味の時間がない」という私にとって「趣味を仕事にしている人」がちょっとうらやましい（実際は大変だということがわかっていても）。

野田　敏之(23)神奈川県

人それぞれ，苦労をしている状況は変わりません。せめて，前向きに考えましょう。あまりごちゃごちゃ考えていると，最終的に生きていくのは大変だ，という身も蓋もない結論になってしまいますよ。

◆12月3日のことでした。学校から帰るとお兄ちゃんが箱に埋もれていました。なにをやっているのかと思ってふと見ると，パー子（EXPERT）が机の上に……ん？　でもなにか違う。X68030，え～っ！　お兄ちゃんたらX68030を買ってきたのね。これで「変態仮面」が遊べるわ。名前は「さん丸子」にしました。でもお兄ちゃんは，

SATURNとPlayStationで遊んでいるの。大丈夫よ，さん丸子。私が遊んであげるからね。
綿引　一代(15)茨城県
そうして，自分の知らぬ間に環境が構築され，妹にX68030を乗っ取られてしまった兄の運命やいかに!?

◆卒論の執筆にLaTeXを使った。東洋系のため，縦書きかつ脚注も最後にまとめてくるので，あまりLaTeXらしさを引き出せずに残念だった。しかし，瀧氏の連載がなければ留年してたかも。
師　茂樹(22)東京都
師さんは無事に卒業できるようですが，瀧氏は無事に進級できるのかなあ。

◆この間，朝学校へ行く途中でガソリンスタンドに寄ってガソリンを入れたときのことです(私は原付で通学している)。ガソリンを入れている最中，ふとテーブルの上にガソリンタンクの蓋があったんですよ。私のは店員が持っているし……どうやら私の前にきた客の原付の蓋を締め忘れたらしいのですが。店員は慌てて前の客を追っかけていった。恐ろしい。
青木　恭一郎(20)東京都
そして，必死に追いかけた店員の目の前で蓋を開けたまま走っていった原付のガソリンに引火して……なあんてね，うふふ。

◆最近，LDプレイヤーが安くなっていますが，ユーザーは多いのでしょうか。僕はあってもいいかな，と思いますが特にほしいとは思いません。理由は，音楽は何回も聴くのですが，映像を何回も見たいとは思わないからです。たまには昔見た映画が見たいなと思うこともありますが，そう頻繁ではないのです。だからビデオもLDもソフトを買うことはありません。このような映像と音楽に対する接し方の違いをもっているのは，僕だけなのでしょうか。皆さんはどう思いますか。
清野　一男(23)山形県
映像は音だけのCDよりも情報量が多いので，頻繁に見ると疲れてしまうからなのか。それとも音よりも映像のほうが，心に残りやすいので見返す必要がないのか。う～む，よくわからん。

◆先日，ジョイスティックを殴って壊してしまった。基板まで割れてしまい，しかたなく新しいものを買いました。壊れたジョイスティックを捨てるのももったいないので，コードとハンダでチクチクやっていたら直ってしまいました。以前も，某対戦格闘ゲーム用の6ボタンスティックを壊してしまい，ハンダでチクチクやっていたことがありました。2台分解してジョイスティックの内部もだいたい理解したので，自家製ジョイスティックでも作ろうかな，と思う今日この頃です。
大平　篤秀(21)東京都
壊され続けるのはかわいそうなので，自作のジョイスティックは，ちょっとやそっとの衝撃では壊れない鋼鉄のボディにしてあげてね。

◆きなこ餅は塩を練り込んである自家製の餅で作ると最高です。正月に僕はハマってしまいました。
竹崎　聞(18)埼玉県
餅に含まれる塩味がきなこの甘さを引き立てるわけですね。いいかもしれない。う，こんなことを考えているとまたおなかがすいてしまう。

◆今年もX68000ユーザーの方々から年賀状をいただきました。わ～い。なかでも某氏のハガキはクジ番号の末尾の下2桁を，わざわざ「68」となっているものを選んで送ってくださるという凝りようで面白かったです。皆変なんですが……ははは。
岩瀬　貴代美(23)福岡県
このような話を聞くと，本当にユーザーはX68000が好きなんだなあと感じます。これからもこんなこだわりをもち続けて，楽しいパソコンライフを送ってくださいね。

◆小さな本屋を見つけました。仕入れが少ないのか，雑誌の並べ方が特徴的です。平積みは平積みなのですが，雑誌のロゴが見えるように段段に重ねられているのです。いい仕事をしていると感じました。
中島　民哉(24)埼玉県
買いにくるお客さんのことをきちんと考えているなんて泣かせますねえ。ところでその本屋にはOh!Xが置いてありました？

◆いやあ，斜めのカッティングのハガキはカッコイイですね（笑）　佐藤　友一郎(21)宮城県
佐藤さんだけかと思って，アンケートハガキを見ていくと，斜めのハガキが結構ありました。ごめんなさい。もう，印刷所の小人さんたらオチャメなんだから。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

売ります

★システムサコムのSCSIボード「SX-68SC」を14,000円で売ります。箱，説明書すべてあり。連絡は往復ハガキでお願いします。〒465　愛知県名古屋市名東区藤ヶ丘市街地住宅5棟902号　飯田　博(26)

★アイ・オー・データ機器の2MバイトRAMボード「PIO-6BE-2M」を10,000円，カラーイメージユニット「CZ-6VT1」を20,000円ぐらい，HAL研のイメージスキャナ「HGS-68」を10,000円ぐらいで売ります。すべて送料込みで，箱，説明書，付属品つきの完動美品です。連絡は往復ハガキでお願いします。〒033　青森県三沢市幸町3-16-13-3-102　松浦　学(39)

★アイ・オー・データ機器の4MバイトRAMボード「PIO-6BE4-4M」を25,000円で，MS-DOSエミュレータ「コンチェルト-X68K」を25,000円以上で売ります。箱，付属品すべて揃っています。連絡は往復ハガキでお願いします。〒343　埼玉県越谷市弥栄町1-105-27　加茂　択也(20)

★X68000 XVI用2Mバイト増設RAMボード「CZ-6BE2B」を25,000円で売ります(送料込み)。連絡は往復ハガキでお願いします。先着順で決めます。〒243-02　神奈川県厚木市下荻野1313-5　萩原　保憲(27)

★東京システムリサーチのX68000 XVI用増設メモリボード「XSIMM10」（増設SIMM 1Mバイト×2つき）を18,000円で売ります（送料込み）。箱，説明書ありです。連絡は往復ハガキでお願いします。〒247　神奈川県横浜市栄区本郷台1-25-18　永井　敏之(24)

買います

★SCSIボード「CZ-6BS1」(説明書，付属品つき，完動品）を12,000円で買います。連絡は往復ハガキでお願いします。〒355-01　埼玉県比企郡吉見町上砂528-1　河上　博仁(18)

★X1turbo用のカラーイメージボード「CZ-8BV2」，カラー熱転写プリンタ「CZ-8PC系」を適価で買います。官製ハガキに売りたい価格を書いて連絡してください。〒465　愛知県名古屋市名東区高針3-203-207　小林　武治(26)

★ローランドの音源モジュール「SC-55」とシャープのMIDIボード「CZ-6BM1A」かシステムサコムのMIDIボード「SX-68MII」をセットで45,000円（送料込み）で買います。説明書，付属品ありでお願いします。連絡は往復ハガキでお願いします。〒799-26　愛媛県松山市福角町625-8　加藤　和人(19)

★コパルのCD-ROMドライブ「CS-CD301X」を15,000円で買います（送料込み）。連絡は往復ハガキでお願いします。〒206　東京都多摩市馬引沢2-13-17　増田　秀樹（28）

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は1月号の内容に関するレポートです。

●CD-ROMドライブ。いままでは，あまり興味の対象とされない感じだったこのSCSI機器は，私の中でX68000にほしい周辺機器の第2位になっています（1位はモデム，3位は増設メモリ）。最近私は一眼レフを購入し，さらにPhotoCDの存在も知りました。で，好きなものを写真に撮り，PhotoCDにすれば，コンピュータで加工したり永久保存したりできます（画像の劣化なしに）。そして，値段も高くなく，意外と役に立つのではと思い始めています。ほかにも効果音CDを買ってくれば，すぐにでもSX-WINDOWが賑やかになるかもしれません（ベル.Xを使用すれば）。悦楽志向のような気がしますが「あったらきっと楽しくなるかもしれない」ものとして，CD-ROMは私にとってはたいへん魅力のある製品です。ドライブの購入も現在検討中です。

大上　幸宏（22）　X68000 PROII　鹿児島県

●1月号の特集のタイトルが「割り切って使うCD-ROM」とありますが，割り切って使うのはなにもX68000だけではないような気がします。CDに収録されたデータを利用するという段階では，どの機種であろうが使用感は同じではないでしょうか。CD-ROMをサポートしている他機種とX68000の違いは「ソフト供給さえもCDで行うかどうか」ぐらいなものでしょう。そう考えるとX68000にはCD-ROMは似合わないのかもしれません。Oh!Xの付録ディスクに代表されるように「フロッピーによいものを強引に詰め込む」といった文化（かな？）が，X68000にはあるように思えます。CDはX68000にとっては広すぎる「ただの空間」なのかもしれませんね。

あと1月号で目についたのが「ANOTHER CG WORLD」です。特に6コマめの言葉には感心させられました。人は太古から「バーチャルリアリティ」していたのですよ，たぶん。だからいろいろな夢に関する分析法や夢を操作する方法を考え出してきたのですね。しかし，夢の威力はすごいものです。おおよそ現実では起こりえない現象が展開されていても，「これは夢だ」と自覚できる人はめったにいないと思います。逆に「バーチャルリアリティ」を追求したゲームをプレイしても，大部分の人が「これはゲームだ」と自覚しながらプレイするでしょう。その夢の威力を考慮したうえで，私は6コマめの言葉にひと言つけ加えさせていただきます。「夢はいちばん手軽で“究極のバーチャルリアリティ”である」と。

中矢　史朗（24）　X68030，X68000 ACE-HD，PC-386

●1月号の特集を読んで，X68000でも読めるCD-ROMが意外に多いことに驚かされました。基本的に一度メモリに取り込んでしまえばどうにでもなるのがデジタルデータなので，機種に依存したデータが増えることはいいことだと思います。新製品紹介にあった「Xellent30」は，FPUつきであること，ソフトウェアで68000←→68030モードに切り替えられることを考えれば，かなりポイントが高い製品だと思います。しかし，ゲームなどでHuman68k ver.2を使っているIPL起動ソフトが使えるのかどうか気がかりです。

進藤　慎一（24）　X68000 EXPERT-HD　青森県

●現在，X68000ユーザーの期待を裏切らないサードパーティの製品「Xellent30」のレポートは，ここ最近のOh!Xの記事の中で最も興味深かった記事です。確かに純粋なX68030に比べると速度的にはそれなりですが，コストパフォーマンスを考えるとなかなかすごいと思います。やはり，10，16MHzオンリーユーザーは，「X68030の魅力はわかっていても，先立つものが」という方も多いでしょうから。

北野　雅利（29）　X68000 EXPERT　大阪府

●1月号の特集では，ファイルフォーマットの解説があるのがいいです。オーディオ関係のSCSIコマンドの解説もいいですね。CD-ROMの導入に関しては心配ないくらいに記述してあるし，導入したってたいしたことはできないよとも書いてあるのだけれど，それ以上にもそれ以下でもないような特集のような気がしました。

「石の言葉，言葉の夢」の「ダブルクリックのなくなる日」が面白かったです。確かになにも知らずにウィンドウシステムを触ったら，ダブルクリックなんて絶対に思いつきませんよね。しかし，ボタンはともかく，アイコンの実行がシングルクリックになってしまったら，非常にうっとうしいことになります。やはり「Macintosh BASIC」のような操作練習プログラムをつけるのが，いちばんいいような気がしますね。

石田　伯仁（21）X68030，MZ-731，PC-8801mkIIMR,PC-E200　神奈川県

●1月号の特集「割り切って使うCD-ROM」は，意外に面白かったというのが第一印象でした。やはり「X68000で使う」ということに徹したための面白さでしょう。CD-ROMというと，絵や音がジャカジャカ勝手に出てきてユーザーはへらへらとマウスをクリックする，というイメージが一般的だと思いますが（私の偏見？），そういうイメージを裏切る質実剛健な記事たちが素敵です。まあ，ほかにやりようがないともいえますが。ただ，どんなデータがどこにあるかということについてもう少し詳しく書かれていないと，興味をもった人でも手を出しにくいのではないでしょうか。そうそう，「TeX入門講座」ってもうおしまいなんですか？　私はもっと突っ込んだ内容を期待していたのですが。そうか「入門」はあくまで入門ということなのでしょうか。でもせっかく入門したのですから「TeX中級講座」でも「TeX登龍門」でも，とにかく次のステップを期待しています。X68000独自のfontmanの事情もあるとよかったですね。あと1月号のサンプルに使われていたフォントはきれいですね。JGフォントでしょうか。こういうこともちょっと気になりました。

矢野　啓介（21）　X68000 XVI　北海道

ごめんなさいのコーナー

2月号　愛読者プレゼント

P.113　キングレコード提供の卓上カレンダーの当選者数が抜けていました。正しい当選者数は2名です。

バグに関するお問い合わせは

☎03(5642)8182(直通)

月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

理論を学び感性を育てよう

▶今月の特集は「SoundEffects」。FM音源による効果音作成，高速フーリエ交換を使った周波数解析などをお届けしました。最近ではPCM音源が手軽に利用できるようになり，リアリティのあるサウンドを創ることが比較的簡単になっています。必要な機材を使えれば，サウンドの加工も思いのままです。しかし，内部的にはどのような処理を経て加工されているのでしょうか。理論を踏まえてこそ，応用が利きます。少しだけ難しいかもしれませんが，がんばって読みこなしていきましょう。

また，Z-MUSICもver.3.0に向けて着々と進化しています。こちらも期待大，というところですね。

▶今年も読者参加の特別企画「言わせてくれなくちゃだワ」のアンケート用紙がやってきました。封筒でお送りしていただくという手間はありますが，読者の皆さんの意見を主張できるチャンスです。日頃いいたくてもいえなかったこと，なんでも好きなことを今月号にはさみ込まれているアンケート用紙にぶつけてください。スペースに書けるだけ書いて送っちゃいましょう。読者の協力がないと成り立たないこの企画，ぜひ皆さんの参加をお待ちしています。

また，カラーイラストも掲載予定ですので，お絵書き職人の読者の皆さん，力作をお待ちしていますよ。

▶４月号では，1994年度Oh!X GAME OF THE YEARの発表が行われます。どんな作品がどんな賞を受賞するのか，非常に楽しみですね。

▶期待のDSPボード「AWESOME-X」は，ソフトウェア関連で遅れが出てしまい，今月号に紹介することができませんでした。楽しみにしていただいた読者の皆様には本当に申しわけありませんでした。製品版が届きしだいレポートしたいと思います。

▶「ハードコア3Dエクスタシー SIDE B」は著者急病のため，「X68000マシン語プログラミング」は，残念ながらお休みとなってしまいました。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「㋢㋮㋴」係

SHIFT・BREAK

▶とにかく忙しい。大学はもう懲りた。これからはVF2にすべてを捧げる。３段が決まれば勝ちが確定のアキラが壮絶。といっても，成功率は1％未満。CRとまではいかなくても権利物くらいの確率かな。そういえば，今年はスキーに行っていない。病気をしていたことだし，しばらくは体を休めることに専念するか。（横）

▶夜道を歩いていたら前を歩いていた女性に，痴漢と間違われて警察に通報された。翌日の夜は交差点で，信号無視の車にはねられて，倒れている間に逃げられた。もう痛くないけどそのときはものすごく痛かった。まったく，冗談じゃないよ。なんか最近ついていないぞ。悪いことしたかなあ。……してるかなぁ。（瀧）

▶４倍速のCD-ROMドライブを購入した。といっても，X68000にではなく，AT互換機の内蔵用だ。さすがに４倍速になると，ほとんどストレスなく使用できる。これでうちのマシンはMPC3対応だ。Reel Magicも買ってきたので，ビデオCDも見られるようになったし。ああっ，マルチメディアという言葉に踊らされてる!?（I.K）

▶引越の際，最後まで段ボールに入れなかったのがX68000と周辺機器一式。引越後，最初に段ボールから取り出したのもやはりX68000とその仲間たち。んー，このへんに自分の生活というものがよく出ているかもしれない。はたしてフライパンや鍋が段ボールから出されて自炊再開できるのはいつの日か（ちなみに引越から１カ月たちました）。（哲）

▶ドラネコに最近悩まされている。布団を干せば小便ひっかけに来るし，車のボンネットで座談会おっぱじめるわで，どーにかしたい。だいたいなぜこんなにネコが出てくるようになったんだ，と調べてみたらある朝，近所の住人の部屋の前に残飯の載った皿が！　餌づけしてんじゃねぇ～。ネコイラズじゃなくてイヌイラズみたいのってないの？（善）

▶ショック。ユナイテッド航空バンコク発東京経由米国往復割引キップが３月で発売中止に。うまく使えば東京米国単純往復より安かったのに。さらにショックなのはこれを誰にいっても，なにがショックなのかわからないらしいこと。そりゃ，米国へのキップをタイに買いに行くなんて，一般人にはわからないかなぁ。マイナーな趣味って悲しい。（で）

▶セガの新作ラリーゲームをカンクネンとオリオールがプレイして（セリカの実車をドライブしている縁で），そのリアルさを絶賛したそうな。私はそれがリップサービスであることを切に願っている。ビデオマガジンでときどき見る彼らの運転は尋常ではない（誰もが驚くはず）。そんなレベルに合ったラリーゲームが遊べるとは思えないのである。（A.T.）

▶映像は真実を映さないし，テレビは何も伝えないことを明らかにした阪神大震災報道。テレビを見るより，もしこいつが倒壊したらと高層マンションを見上げて想像するほうがよほど高いリアリティ。マスコミヘリのローター音のせいで瓦礫の下から洩れるかすかな声が聞えなかったという話も聞くにつれTVからまたもや遠ざかってしまうのだ。（K）

▶昔，レコード（CD）でミリオンセラーを記録すればそれなりに世間の人も知っていたような気がする。ところが，去年は20曲近くのミリオンセラーが生まれたのに，曲名を挙げられない人が多いだろう（読者はそうでもないかな）。それは人口が増えたからっていうのは冗談で，売れてる世代が狭いから。要はマーケティングがうまいわけだけど……。（高）

▶農薬バリバリで一時期問題となった烏龍茶のメーカーはどこなんだろう。裁判沙汰にまでなって，結局メーカー名を公表するという話だったのだが……。おかげで，ここしばらく烏龍茶が怖くて飲めない状況が続いている。タバコをガンガンに吸って，徹夜バリバリの不健康人間が，こんなことを気にしてもしょうがないかもしれないけど。（J）

▶家賃の相場も下がってるし，もう少し広いとこへ引っ越しをしたいと思ってたのだが，いつまでたっても更新の案内がこない。確か通知は契約切れの３カ月前のはずだが，２年契約じゃなかったっけ……。で，今月の家賃を持ってくと「来月でおしまいだから，じゃあ更新ですね」といわれてしまった。いまから探すのも辛いが意地でも越すべきか？（U）

▶救助隊も消防車もぜんぜんこないのに，どうして報道関係者はあんなにたくさんやってくるのだろう。被害の少ない周辺の自治体が水もない避難所の人々を保護することはできなかったのだろうか。場所によっては歩いて行けるはずなのだが，そういった情報もなかったようだ。休暇が取れたら帰りたいんだけどね。（神戸，芦屋，西宮で育ったT）

microOdyssey

巷でビタミンCやβ-カロチン，カルシウムなどが入ったお菓子のようなビタミン食品をよく見かける。その形状はともかく色が似ている。ビタミンCが多く含まれるものは淡い黄色，β-カロチンなら橙色，カルシウムなら白色といったところだ。要はそれぞれの栄養素が含まれる代表的な食品，レモンやニンジン，牛乳をイメージさせるからだろう。なぜか，その色に安心して，その食品を食べたような気になる(ちょっと大げさか)。もし，これが黒い丸薬だったら，また別の感覚があるかもしれない。

あるひとつの実験がある。まず，ラベルをつけない4個のコーヒー缶(濃い茶，赤，青，黄色に塗ったもの)を用意する。それを4カ所のブースに置く。そして，ひとつのコーヒーメーカーから入れたコーヒーをカップに注ぎ，缶のそばに置いておく。被験者はそれぞれのブースに入ってコーヒーを味わう。そこで，コーヒーの味についての質問に答えてもらう。すると，濃い茶色の缶を見た人の約75%は味が極めて濃いと感じ，黄色の缶を見た人の約85%は味が薄いと感じたそうだ。このように色が味覚にたいして影響を与えたのだ。

色が日常の生活に影響を与えるもっともありふれた例は服の色だろう。日射しが強く暑いときは白っぽい服を着て，寒いときなら黒っぽい服を着ることが多い。こんなことは誰でも知っている。では，着ている服が見えなかったら，それも室内だったらどうだろうか？

衣服に関してではないが，こんな実験があった。赤と青の壁で囲われた2つの部屋を用意して，状態をどちらも同じに保つ。そこに目隠しをした被験者に入ってもらった。すると，赤い部屋では皮膚温度が上昇し，血圧が高くなり呼吸数も増えた。青い部屋では血圧が下がり呼吸数も減った。つまり，視覚でなく皮膚で色の変化を感じとったのだ。

このように色は人間にいろいろと影響を与えている。部屋の色はもちろん，そこに置いてあるものの色も生活していく上で重要なはずである。電化製品で白ものとか黒ものとかいわれるが，最近では一部のもので選べる色に変化が見られる。あえて区別すれば黒もののパソコンの色を見てみると，X680x0は黒がほとんどだが，全体を見ればオフィスグレーなどが主流だ。特に日本の場合，オフィスに導入されている台数のほうが多いのだから，仕方のないところか。でも，パーソナルがうたわれ，実生活に入り込むなら，色くらいは自分で好きなものが選びたい(その前に改善すべき点はいろいろあるだろうが)。その点，X1は選択肢があった分，その色がどうであろうと，好きな機種だった。

個人的には，フレームが木製(たんに木目調だと安っぽい)でWindowの壁紙は木目や畳なんかを背景にして……量産は絶対無理だな。　(高)

●

兵庫県南部地震により，お亡くなりになられた方々のご冥福をお祈りするとともに，被災されました皆様方に心からお見舞い申し上げます。

ソフトバンク株式会社

1995年4月号3月18日(土)発売

特集　ゲームの攻略，請負います

・ザインの軌跡　・2周目からの快楽
・いけずなボスの倒し方　・これぞハイテクプレイ!?

1994年度GAME OF THE YEAR発表

試用レポート　MJ-5000C

全機種共通システム
S-OSねちねち入門(1)

バックナンバー常備店

地域	地区	店名・電話
東京	神保町	三省堂神田本店5F 03(3233)3312
	〃	書泉ブックマートBI 03(3294)0011
	〃	書泉グランデ5F 03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(3257)2660
	八重洲	八重洲ブックセンター3F 03(3281)1811
	新宿	紀伊国屋書店本店 03(3354)0131
	高田馬場	未来堂書店 03(3209)0656
	渋谷	大盛堂書店 03(3463)0511
	池袋	旭屋書店池袋店 03(3986)0311
	八王子	くまざわ書店八王子本店 0426(25)1201
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 043(224)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X　3月号
■1995年3月1日発行　定価680円(本体660円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部　☎03(5642)8122
販売局　☎03(5642)8100 FAX 03(5641)3424
広告局　☎03(5642)8111
■印刷　凸版印刷株式会社

満開の電子ちゃん

作・え　岡村祭

81号（1/18発送）は、IVM用BMPコードリソースとか、ザウルス活用ツールとか、「Xellent30」「XL/Image」レポートなど。「魔法大作戦」10MHz用パッチもあるでよ。

谷本和生（広島県）

電脳倶楽部も最近はいろんな場所に進出してタケル・ニフティサーブなどでも購入できます。私は創刊1年目からの購読で、毎月20日前後は郵便ポストが気になります。電クラで時間を共有するのは習慣になっていて、もうやめられません。
また、こんなに長期間続いているディスクマガジンは電クラだけでしょう。それは、読者の皆さんの参加によって素晴らしいものを創り出しているからです。電クラはパソコン文化を創っていると思います。Oh！Xの読者の皆さんも是非参加して下さい。

マイコン専門ショップ

P&A

SHARPエキスパートショップ

今が購入のチャンス!

2/17〜3/17

X68030お買い得セット

(クレジット表:送料・消費税込み)

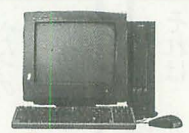
●CZ-500C
●CZ-608D(B)

定価¥492,800
P&A超特価
¥302,000

クレジット表	12回	27,400	24回	14,400
36回	10,000	48回	7,800	60回	6,500

●CZ-510C
●CZ-608D(B)

定価¥582,800
P&A超特価
¥401,000

クレジット表	12回	36,300	24回	19,100
36回	13,200	48回	10,300	60回	8,600

●CZ-300C
●CZ-608D(B)

定価¥482,800
P&A超特価
¥302,000

クレジット表	12回	27,400	24回	14,400
36回	10,000	48回	7,800	60回	6,500

●CZ-310C
●CZ-608D(B)

定価¥572,800
P&A超特価
¥396,000

クレジット表	12回	36,000	24回	18,900
36回	13,000	48回	10,200	60回	8,500

■モニター変更の場合
●CZ-615D(チューナー付)に変更の場合¥56,000
●CZ-621D(B)…………に変更の場合¥64,000
加算して下さい。

決算大処分セール 旧シリーズ今が買いどき!!

(送料¥2,000・消費税別)(クレジット表:送料・消費税込み)

X68000 Compact XVI

●CZ-674C-H
●CZ-608D(B)

定価¥392,800
P&A超特価**¥145,000**

12回	13,200	24回	7,000	36回	4,800	48回	3,800	60回	3,100

●CZ-674C-H
●CZ-608D(B)
●CZ-6FD5

定価¥492,600
P&A超特価**¥193,000**

12回	17,600	24回	9,200	36回	6,400	48回	5,000	60回	4,200

決算大処分セール 旧シリーズ今が買いどき!! (送料¥1,000 消費税別) 単品、限定

⊙PROII-HD

●CZ-663C
(ハードディスク40MB内蔵)
P&A超特価
¥49,800

⊙PRO

●CZ-652C
P&A超特価**¥46,800**

⊙PROII

●CZ-653C
P&A超特価**¥47,800**

⊙CompactXVI

●CZ-674C
P&A超特価
¥79,800

MIDIセット

●MC-6600(SNE)
●SX-68MII(システムサコム)
●MIDIケーブル
特価¥48,500

●SC-55MKII(ローランド)
●SX-68MII(システムサコム)
●MIDIケーブル
特価¥70,800

(SC-88に変更の場合¥17,000加算して下さい。)

単品
●MC-6600(SNE)…………特価¥34,800
●SC-55MKII(ローランド)・特価¥56,800
●SC-88(ローランド)……特価¥73,500
●05R/W(KORG、加賀電子)・特価¥49,800

スピーカー

●MS-3000(SNE)……特価¥11,500
●SC-C55(AIWA)………特価¥ 5,980

ALTEC ACS300
特価¥37,000

ALTEC ACS100
特価¥16,000

SHARP CP-A5-B
特価¥9,400

X68000/68030用 メモリボード (送料¥700・消費税別)

■I/Oデータ
●SH-5BE4-8M(30用)……特価¥39,500
●SH-6BE1-1ME(600C用)…特価¥10,200
●PIO-6BE1-AE(ACE/PRO/PROII用)・特価¥10,200
●PIO-6BE2-2ME(拡張スロット用)・特価¥21,000
●PIO-6BE4-4ME(〃)・特価¥35,300

■シャープ
●CZ-5BE4(30用)………特価¥39,800
●CZ-5ME4(5BE4用増設)・特価¥36,500
●CZ-6BE2A(XVI用)……特価¥38,900
●CZ-6BE2B(XVI、674C増設)特価¥37,500
●CZ-6BE2D(674C用)…特価¥20,500

モデム&FAXモデム (送料¥1,000)

〈アイワ〉
●PV-BF144(ボックス型)
…………………………特価¥17,000
●PV-AFV144(液晶パネル、ボックス型)
…………………………特価¥26,800
●PV-PFV144(ポケット型)
…………………………特価¥22,800

〈オムロン〉
●ME1414BII(ボックス型)………特価¥17,000
●MD144XT10V(ボックス型)……特価¥34,000

〈マイクロコア〉
●MC14400FX(W)(ボックス型)…特価¥23,000
●MC24FC5(W)(ポケット型)……特価¥20,000

MO&CD-ROM (送料¥1,000)

■CS-M230PA(コパル)
光磁気ディスク(X68000用)
●ケーブル付
特価¥102,000

■LMO-FMX330TS
(ロジテック)●ケーブル付
定価¥168,000
特価¥97,000

■MO (ケーブル別売)
●UL-312E-S(緑電子)………特価¥62,000
●MO-120S(ICM)……………特価¥88,000
●MO-230S(〃)……………特価¥110,000
●LMO-340(ロジテック)……特価¥52,300
●LMO-400(〃)………特価¥78,800

■CD-ROM
⊙CDS-E(メルコ)(トレイ、2.4倍速、ソニー)……特価¥23,500
⊙SCD-400(ロジテック)(キャディー、4倍速、東芝)・特価¥36,500
⊙ECD-550(エレコム)(キャディー、4倍速、東芝)・・特価¥44,800
※Driver+SCSIケーブル……………特価¥7,300

東京システムリサーチ製(X SIMM)

(送料¥700・消費税別)

(X SIMM VI)
⊙XVIシリーズ専用SIMM増設式メモリボード
●X SIMM VI(634C用)…定価¥16,500➡特価¥13,000
●X SIMM VIc(674C用)・定価¥16,500➡特価¥13,000
⊙増設SIMMメモリ(72PIN)
●4MB(70ns)……………………特価¥11,800
●8MB(70ns)……………………特価¥27,800
●4MB(60ns、24MHz以上用)………特価¥16,500
●8MB(60ns、24MHz以上用)………特価¥28,000

●6MB(60ns、メーカー純正品)……特価¥27,800
(X SIMM 10)⊙SIMM増設式メモリボード
●X SIMM 10………定価¥18,000➡特価¥15,700
⊙増設SIMMメモリ●1MB×2……特価¥ 9,000
●4MB×2……特価¥30,000
●10MB例 X SIMM10+1MB×2+4MB×2・・¥54,700

X68000/68030専用ハードディスク (送料¥1,000・消費税別)

外付

■ジェフ
⊙GF-340(330MB、13ms)…特価¥29,800
⊙GF-540(520MB、12ms)…特価¥42,000
⊙GF-1000(1060MB、9ms)・特価¥76,000

■ロジテック
⊙SHD-B340NU(340MB、12ms)
(ケーブル、ターミネータ付)…特価¥35,800
⊙SHD-B540U(540MB、10.5ms、256K)
(ケーブル、ターミネータ付)・特価¥49,800

■モッキンバード
⊙HD-M350(350MB、14ms、256K)・特価¥35,800
⊙HD-K520(520MB、12ms、240K)・特価¥45,500

内蔵

■CZ-500C/300C専用
⊙CZ-5H08(80MB/23ms)
…………………………定価¥ 98,000▶特価¥71,800
⊙CZ-5H16(160MB/18ms)
…………………………定価¥135,000▶特価¥99,500

注目!!夏のボーナス一括払い手数料(金利)無料(平成7年3月末/4月末/5月末/6月末/7月末のいずれかをご指定下さい。)

●価格は変動します。ご注文の際は必ずお電話で価格と在庫をご確認下さい。●本広告に掲載の商品には送料及び消費税は含まれておりません。

X68000XVI対応 MPUアクセラレータ
あなたのXVIを030にグレードアップ

Xellent30（東京システムリサーチ 定価¥59,800）

特価¥46,500

（●MPU交換に付き、保証（メーカー、当社）は付きませんのでご承知下さい。）

《業界No.1の"P&Aメンテナンスサポート"》

最高の保証システム

①業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証!! ※一部商品は除きます。）
②中古パソコンの1年間保証（※モニター・プリンター6ヶ月間保証!!）
③初期不良交換期間3ヶ月（※新品商品に限らせていただきます。）
④永久買取保証
⑤配達日の指定OK//（土曜・日曜・祭日もOK!!）
⑥夜間配達もOK//（※PM6:00〜PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利//
③月々の支払いは¥1,000より
④9ヶ月先からのスキップ払いOK//
⑤84回までの分割、ボーナス併用OK//
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK//
⑨現金一括支払いOK//
⑩商品到着払いOK//（代引き手数料が必要になります。10万円まで900円）
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向け リースシステム
業務に最適なシステムを構築します。
損金処理が可能なリース契約をどうぞ。

周辺機器コーナー（送料¥1,000・消費税別）

カラーイメージスキャナ
■JX-330X
定価¥178,000
特価¥118,000

カラーイメージジェット

■IO-735X-B
定価¥248,000
特価¥128,000

ビデオスキャナー
■CZ-6VS1
定価¥178,000
特価¥135,000

FDD（5インチ×2基）

■CZ-6FD5
定価¥99,800
P&A超特価
¥49,800

プリンター（ケーブル用紙付）
- MJ-500V2（エプソン）……特価¥31,300
- MJ-1000V2（〃）……特価¥51,300
- MJ-700V2C（〃）……特価¥64,800
- BJ-220JCII（キヤノン）…特価¥53,400
- BJ-10V Lite（〃）…特価¥27,800
- BJ-15V PRO（〃）…特価¥39,700
- LBP-A404GII（〃）…特価¥87,300
- BJC-600J（〃）…特価¥66,000
- BJC-400J（〃）…特価¥54,300

ペン&タブレット

■Drawing Slate（NS・カルコンプ）
- 31090SER（6×9）定価¥74,800▶特価¥58,500
- 31120SER（A4）定価¥79,800▶特価¥63,000
- 31180SER（A3）定価¥99,800▶特価¥78,500

- CZ-6BV1………定価¥21,000▶特価¥15,900
- CZ-8NM3………定価¥9,800▶特価¥7,200
- SH-6BF1………定価¥49,800▶特価¥36,500
- CZ-6BP1………定価¥79,800▶特価¥57,000
- CZ-6BS1………定価¥29,800▶特価¥21,500
- CZ-8NJ2（限定）…定価¥23,800▶特価¥13,800
- CZ-6CS1（674C用）・定価¥12,000▶特価¥8,900
- CZ-6CR1（RGBケーブル）・定価¥4,500▶特価¥3,600
- CZ6CT1（テレビコントロール）・定価¥5,500▶特価¥4,400
- CZ-6BP2………定価¥45,800▶特価¥33,300
- CZ-5MP1（X68030用）・定価¥54,800▶特価¥42,000

送料¥700・消費税別

■システムサコムボード
- SX-68MII（MIDI）定価¥19,800 特価¥13,500
- SX-68SC（SCSI）定価¥26,800 特価¥17,500

X68000用ソフトコーナー（送料¥700・消費税別）

〈シャープ〉
- CYBERNOTE PRO68K（CZ-243BSD）……特価¥15,000
- MUSIC PRO68K（MIDI）（CZ-247MSD）……特価¥20,500
- CANVAS PRO68K（CZ-249GSD）特価¥22,000
- Easypaint SX-68K（CZ-263GWD）……特価¥9,800
- Easy draw SX-68K（CZ-264GWD）・特価¥15,300
- New Print Shop Ver.2.0（CZ-265HSD）……特価¥15,400
- Press Conductor PRO68K（CZ-266BSD）……特価¥22,000
- CHART PRO68K（CZ-267BSD）・・特価¥29,800
- EG-Word（CZ-271BWD）……特価¥44,900
- Communication SX68K（CZ-272CWD）……特価¥14,500
- Datacalc SX-68K（CZ-273BWD）……特価¥44,000
- MUSIC SX68K（CZ-274MWD）・・特価¥29,300
- SOUND SX68K（CZ-275MWD）・・特価¥11,500
- フォント・アンド・ロゴデザインツール SX-68K（CZ-282BWD）……特価¥22,000
- BUSINESS PRO68K（CZ-286BSD）……特価¥20,500
- 開発キット（work room）（CZ-288LWD）……特価¥29,700
- 開発キット用ツール集（CZ-289TWD）……特価¥9,600
- SX-WINDOWディスクアクセサリー集（CZ-290TWD）……特価¥11,500
- XDTP-SX68K（CZ-291BWD）……特価¥26,900
- C-Compiler PRO68K Ver.2.1（CZ-295LSD）NEW KIT……特価¥32,500
- SX-WINDOWS Ver.3.1（CZ-296SS/SSC）……特価¥17,600

〈マイクロウェア〉
- OS-9/X68030 V.2.4.5……特価¥19,900
- X-WINDOWS V.11 R5……特価¥25,500
- Technical Tool Kit V.2.4.5……特価¥17,000
- Ultra Cアンド Professional Pack V.1.1……特価¥38,000
- Video PC for X680X0……特価¥57,000

〈計測技研〉
- Free Software Selection Vol.2……特価¥4,800
- Double Bookin……特価¥9,600
- CD-ROM Driver V.2.0……特価¥3,800
- シャーペンワープロパック……特価¥5,400

〈その他〉
- F-Card V5 for X68K（クレスト）……特価¥9,600
- F-Calc for X68K（クレスト）……特価¥11,000
- たーみのる2（SPS）……特価¥13,000
- MU-1GS（サンワード）……特価¥21,000
- マチエール Ver.2.0（サンワード）……特価¥28,800
- Z's STAFF PRO68K Ver.3.0（ツァイト）……特価¥37,500
- Z's TRIPHONYデジタルクラフト（ツァイト）……特価¥27,000
- XL/Image（IMAGICAテクノシステム）……特価¥46,000

全国通販 ★頭金なし！★即日発送

- ●お近くの方はお立寄り下さい。専門係員が説明いたします。
- ●本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
- ●ビジネスソフト定価の20%引きOK！TELください。

P&A特選 今月の中古特選品

単品
- CZ-500CB ¥175,000

- CZ-623C ●68000専用モニター付 ¥96,000

- CZ-653C ●68000専用モニター付 ¥77,000

新品限定
- CZ-652C ……¥46,800
- CZ-653C ……¥47,800
- CZ-663C ……¥49,800

- CZ-600C・・¥40,000
- CZ-601C・・¥40,000
- CZ-611C・・¥45,000
- CZ-652C・・¥39,800
- CZ-612C・・¥60,000
- CZ-603C・・¥53,000
- CZ-653C・・¥41,000
- CZ-612C・・¥65,000
- CZ-623C・・¥75,000
- CZ-674C・・¥59,800
- CZ-634C・・¥110,000
- CZ-644C・・¥145,000

※上記は単品価格、モニター別売。

高額買取り（新品もOK）格安販売

■まずはお電話下さい。下取り専用 買取り電話▶ ☎03-3651-1884 FAX.03-3651-0141

買取り価格…完動品・箱／マニュアル／付属品の価格です。中古販売…1年間保証付。

- ●下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
- ●買取りの場合…現品が着き次第、3日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。

- ●最新の在庫情報・価格はお電話にてお問い合せください。
- ●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
- ●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
- ●本商品の掲載の商品の価格については、消費税は、含まれておりません。
- ●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

P&Aオリジナル特選パソコンラック&OAチェアー（消費税込み）（送料無料、離島を除く）

①¥10,815（2段別々使用OK）

※キャスター付、4段、17"モニターOK、色（グレー）。
※上から2番目棚板移動可能。

②¥12,360（マウステーブルスライドOK）

※キャスター付、4段、17"モニターOK、色（グレー）。
※スライドマウステーブル、中棚板は2段階移動可能。

①¥4,944
- ●布張り 色（グレー）
- ●ガス圧シリンダー

②¥6,283
- ●肘付
- ●布張り 色（グレー）
- ●ガス圧シリンダー

※ラック、チェアー持ち帰り可能です。ご来店下さい。

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回〜84回払いまで出来ます。但し、1回のお支払い額は¥1,000円以上。

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。（電信扱いでお振込み下さい。）

〔振込先〕さくら銀行 新小岩支店
当座預金 2408626 （株）ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.6	3.0	4.2	4.89	6.5	10.0	14.3	18.9	24.3	31.8

（※車でお越しの場合は北海道拓殖BK前の新小岩駐車場をご利用下さい。）

P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間：AM10：00〜PM7：00 日・祭：AM10：00〜PM6：00
☎03-3651-0148（代）
●定休日／毎週水曜日
FAX.03-3651-0141
MAC/DOS Vフロア ☎03-3655-4454

※お支払いは、便利な商品到着払い〈手数料（10万円まで900円）要〉をご利用下さい。

ソフトバンクの新刊

SOFTBANK CD•BOOK

メガドライブで好評の「ラングリッサーII」がドラマCDになる!

メモリアルドラマCD&ファンブック
ラングリッサーII

© NCS

秘剣“ラングリッサー”をめぐって、
激しい戦いの幕が開こうとしている。
エルウィンとレオンとの宿命の戦いが、いま始まる!!
そしてファンブックでは、うるし原智志氏デザインによるキャラクターの魅力を徹底紹介&「ラングリッサー」シリーズの歴史を検証。そのほか、特別インタビューやメイキングなどを収録。

定価3,800円(税込)

豪華声優陣を起用!

【CAST】

エルウィン……草尾 毅
シェリー………横山智佐
リアナ…………國府田マリ子
エリザ…………林原めぐみ
エグベルト……青野 武
ナレーター……銀河万丈
ジェシカ………藩 恵子
ヘイン…………山口勝平
レオン…………置鮎龍太郎
レアード………堀川 亮
バルガス………郷里大輔

RPG幻想事典 アイテムガイド

ヘッドルーム 編著　A5判・予価1,800円

3月上旬発売予定!

西洋ファンタジーに登場する物を中心に、武器や防具などのキャラクターが装備するアイテムについて解説します。いままでの解説本ではあまり触れられていなかった、どんな人が、なんのため、どのように使っていたのかを明らかにするため、歴史や由来、具体的な使用方法などを、イラストを使いながらわかりやすく解説します。また、実際に使われた物だけでなく神話・ゲームなどの有名な魔法の武器についても、詳しく解説します。

RPG幻想事典シリーズ◆好評発売中!

逆引きモンスターガイド東洋編
ヘッドルーム 編著
定価1,800円

逆引きモンスターガイド西洋編
ヘッドルーム 編著
定価1,800円

戦士たちの時代 司史生/坂東いるか 共著 定価1,800円

チャンバラ英雄伝 柳川/高井/横山 共著 定価1,800円

RPG幻想事典・日本編 飯島健男 監修 定価1,860円

RPG幻想事典 早川浩 著 定価1,550円

●定価は税込みです　●お近くの書店でお求め下さい

ソフトバンク株式会社/出版事業部 SOFT BANK
販売局 TEL.03-5642-8101

当社製品については、サポートセンター(0286)27-1829までFAXでお問い合わせください。

CD-ROM Driver 2.0 バージョンアップのお知らせ

ご好評いただいておりますCD-ROM Driverですが、このたびVer.2.0からVer.2.1へのバージョンアップサービスを行うことになりました。
新バージョンでは、メルコのCDS-Eに完全対応したほか、他のドライブでもより安定して動作します。
付属のフリーソフトウェアもバージョンが上がっています。
バージョンアップの方法は以下の2つからお選びください。

[1] パソコン通信を利用したバージョンアップ
　バージョンアップ差分をTECOSYS-3(0286-51-1430)で配布しますので、モデムをお持ちの方はこちらにアクセスして、ダウンロードしていただけます。費用は電話代だけ。

[2] 郵送によるバージョンアップ
　モデムをお持ちでない方は、「270円切手を貼った返信用封筒」を同封の上、「CD-ROM Driver Ver2.0」のマスターディスクを下記住所まで郵送してください。折り返し、Ver2.1のマスターディスクをお送りいたします。
〒320　栃木県宇都宮市京町11-18 OYAMAビル2F
(株)計測技研　CD-ROM Driverバージョンアップ係

好評発売中！ SXパワーアップ委員会 シャーペンワープロパック

標準価格 ¥6,800

SXパワーアップ委員会シリーズ第1弾は、シャーペンをさらに強化する「シャーペンワープロパック」です。
シャーペンワープロパックをインストールすることによって、シャーペンが限りなくワープロに近い存在へとパワーアップします。
文字の回転や各種タブ、インデントなど、最新ワープロソフトにも負けない表現力を追加するほか、文系ユーザー待望の縦書き表示、縦書きインライン入力もサポート。それでいて、従来通りの軽快さもそのまま継承しています。

●動作環境
- SX-WINDOW Ver3.1以上
- 空きメモリ300KB程度

●シャーペンに追加される主な機能
- 縦書き入力
- 文字の回転
- ルーラ(定規)の表示
- 各種タブ(均等割付など)およびインデントの設定*
- 各種禁則処理(追い込み均等など)*
- 行揃えの拡張*
- 段組み印刷

*:パラグラフごとに設定可能

●プログラマ向け機能も充実
- 編集中のソースをコンパイルする等、マクロ機能を強化

●付録
- シャーペン外部コマンド開発キット(ライブラリおよびリファレンス)
- IFM ver 4.0

強化されたカスタマイズ機能でさらに強力に…!!

68040搭載アクセラレータ X68040turbo

標準価格 ¥98,000
ヒートシンク別売 ¥1,000

040turboは、68040を搭載したX68030(5インチタイプ)専用のアクセラレータです。040turboを装着することで得られるパフォーマンスは、従来の2〜3倍！ 計算、特に浮動小数点演算中心のソフトならば、さらにそれ以上の高速化も望めます。
詳しくはソフトバンク刊「X68040turbo〜A Story of Makeing "After X68030"〜」(BEEPs著)をご覧ください。
040turboは当社のショップBASIC-HOUSEでの直販、および通販でのみお買い求めいただけます。ご注文いただいてからしばらくお待ちいただく場合もありますので、お早めにご注文ください。

SX-WINDOW用CD-ROM辞書検索ソフト SX広辞苑《EPWING対応版》

標準価格 ¥19,800
岩波書店「広辞苑第4版」CD-ROM版 バンドルセット ¥43,800

●SX広辞苑《EPWING対応版》の特長
- 豊富でパワフルな検索方法により、必要な情報をすばやくピックアップ。
- 使う側に立って操作系をリニューアル。さらに簡単に、さらに鋭く作業を行なえます。
- 広辞苑の最新版である第4版をもとにしたCD-ROMを使用するので、よりコンテンポラリーなキーワードにアクセス可能です。
- SX-WINDOW上で動作するので記事の参照や引用がとても簡単。シャーペンやEGWordと組み合わせて活用できます。（ただし、広辞苑では大量の引用は禁止されています）
- シャーペンと融合して語句の検索を行なうシャーペン用外部コマンド"LightWing.X"を同梱。複雑な検索を行なう場合はSX広辞苑.Xを、普段よく使う単純な検索にはLightWing.Xを、という使い分けも可能です。
- 広辞苑第4版CD-ROM版と同様に、EPWING(V1)規約にもとづいたCD-ROMタイトルなら、ほとんどのCD-ROMの内容を検索できます。

●動作環境
- SX-WINDOW 3.0以上
- SX-WINDOW動作中の空きメモリとして1MB以上を推奨
- CD-ROMドライブ(CD-ROM Driver Ver2.0が付属するので、CD-ROM Driverを別途お買い上げいただく必要はありません。CD-ROM Driverのマニュアルや添付ソフト等は付属しません)

X680x0用Ethernet接続パック EthernetStarterPack/X680x0

期間限定特価 ¥78,000

ESP/Xは、Ethernetアダプタ「Ether+」と、TCP/IPドライバ、そして基本的なアプリケーションからなるパッケージです。
特価期間は2月末日までです。ご注文はお早めに。

- Ether+(米コンパチブルシステムズ社製)
  SCSIインターフェースを介してEthernetとX680x0を接続するためのハードウェアです。
  ※10BASE-2対応モデル・10BASE-T対応モデルの2種類があります。
- TCP/IPドライバ
  X680x0でTCP/IPをサポートするドライバ。ソケットも利用可能です。
- 基本的なアプリケーション
  ftp、telnet(いずれもクライアント)等、基本的なアプリケーションを標準添付。ドライバを活用するためのライブラリも付属します。

●動作環境
- Human68k ver3.0以上
- メモリ常駐量500KB前後
- SCSIインターフェース内蔵機種以外はSCSIボードが必要

※NetWareには未対応です。

X680x0用フリーソフトウェア集CD-ROM

発売中 FreeSoftwareSelection Vol.2　標準価格 ¥6,000

お求めはお近くのパソコンショップ、または当社通販部(TEL:0286-22-9811)へお申し込みください。
通販ご希望の方は、ソフト代金＋送料1,000に消費税を加え、ご住所・お名前・電話番号・商品名を明記した紙を同封の上、現金封筒でお申し込みください。

低金利クレジット　通信販売送料 全国一律￥1,000 長期クレジット可能　※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップBASIC HOUSE 本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

サポートネットTECOSYS-3 24時間稼動中！(0286)51-1430 (9600bps MNP5)

セクシーでパワフルな女子プロを制覇しろ！

18禁版

カードバトルにプロレスを融合させた、「レッスルエンジェルス」シリーズ。いよいよ最大のヒット作「レッスルエンジェルススペシャル」が登場です。さまざまなイベントの選択によって運命が変わる、マルチシナリオ・マルチエンディング。プロレス技数、カテゴリーが増加して、レスラーの個性もパワーアップ。そして、「恐怖の水着はぎデスマッチ」もパワーアップして復活！18禁だから、そのセクシー度はもうケタ違い！待望のX68000移植完成！明日のトップイベンターを目指すのだ！

機能アップ！

- オリジナルオープニングを収録
- 画面のレイアウトを変更
- エキジビションモードグラフィック描き直し
- 256色モードと16色モードを搭載
- サウンドも明るめに変更
- AD-PCMによる効果音
- ディスクアクセスを最少に抑える設計

このソフトは、全国のパソコンショップで、パッケージ版で販売いたします。TAKERUでは販売致しません。TAKERU事務局では通信販売はいたしませんので、悪しからずご了承下さい。

対応機種：X68000/X68030
要メモリ2Mバイト
（ハードディスク対応）
制作：グレイト

¥8,800（税別）

三國志

知力の極限に挑む、君主、武将、軍師の膨大なデータ。小説よりリアルと、名作の誉れ高い中国統一ゲーム。この歴史的な傑作シリーズはどのようにして始まったのか?SLGファンなら絶対に見逃せない!!

制作/光栄
対応機種/X68000（30不可） **¥5,200**

太閤立志伝

裸一貫の足軽頭から身を興し、関白にまで登り詰めた男・木下藤吉郎(豊臣秀吉)。草履を温めたエピソード・奇跡の墨俣一夜城など、数々の逸話を持つ男の一生を再現する、リコエイションゲームの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**

ファランクス

デカキャラ・派手め演出の横スクロールオアワーシューティング。拡大・回転・縮小・多関節・半透明・ラスタースクロール・MIDIと、各種要素がいっぱい詰まってます。

制作／ズーム
対応機種／X68000（30不可） **¥2,500**

三國志 II

登場人物350余名、最大11人まで同時プレイ可能、6編のマルチシナリオ方式、埋伏の毒・駆虎呑狼等のユニークな計略要素導入、さらに深みを増した外交・HEX戦など、まさに名作!カシオペアの向谷　実のBGMも話題に。

制作/光栄
対応機種/X68000（30不可） **¥4,900**

蒼き狼と白き牝鹿　元朝秘史

光栄歴史三部作の一角を成す、草原の英雄チンギス・ハーン。稀代のスケールと空前絶後の迫力で、一代帝国を築き上げた男の豪快な一生を見事に再現!熱いシミュレーションの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**

A列車で行こうII

かの「A列車」シリーズの第2弾。パズル的要素がアツクなる!鉄道会社社長の立場で、線路の敷設・撤去を行い、ワールドワイドにマップを発展させていこう。

制作／アートディンク
対応機種／X68000（30不可） **¥3,800**

大航海時代

リコエイションゲームシリーズの傑作。毎回違った展開が楽しめるイベントジェネレーティングシステム。帆船の特徴が活かされたHEX戦。失われたロマンを求めて、冒険者たちの航海の旅が始まる。

制作/光栄
対応機種/X68000（30可） **¥3,400**

ロイヤルブラッド

新シリーズ「イマジネイションゲーム」のデビュー作。イシュメリアという架空の島国を舞台にした、幻想世界のシミュレーションゲームだ。あなたは独立貴族のひとりとなり、領主達が持っている6つの宝石を集め、イシュメリアの新王となれ!

制作/光栄
対応機種/X68000（30可） **¥2,700**

A III（A列車で行こう3）

さらにワイドに、さらに完成度の増した、世界レベルヒットの第3弾。世にA.IIIブームを巻き起こしたことで、記憶に新しい超有名作、ついに文庫に登場！

制作／アートディンク
対応機種／X68000（30可） **¥3,800**

維新の嵐

坂本龍馬が、西郷隆盛が、吉田松蔭が日本を憂い、改革を目指して奮い立つ!幕末の志士の個性を際だたせる緻密なパラメータ。出会いの楽しさ、駆け引きを楽しむ新システム。強力な機能で、維新を操れ!

制作/光栄
対応機種/X68000（30不可） **¥3,400**

ヨーロッパ戦線

戦乱のヨーロッパ。砂塵の彼方から迫り来る黒い車体は、敵か味方か?次々に飛び込んでくる情報、時事刻々と変わる戦局。多彩な兵器やユニット、人間的要素を重視した各種パラメータ。WWIIシリーズ第2弾。勝利の旗を手に入れろ!

制作/光栄
対応機種/X68000（30可） **¥4,500**

栄冠は君に

高校野球シミュレーションシリーズの、記念すべき第1作。全国制覇を達成するには、3990校の頂点に立たなければならない。感動の優勝セレモニーを、果たして見ることが出来るか！？

制作／アートディンク
対応機種／X68000 **¥3,800**

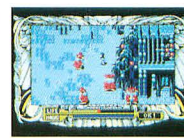
信長の野望　戦国群雄伝

400余名の群雄が割拠する下剋上の乱世。配下の羽柴秀吉、柴田勝家を個性豊かな武将たちを、思いのままに操って、戦雲たなびく戦場へ、天下分け目の決戦に臨む！光栄の代表作「信長の野望」シリーズの傑作！

制作/光栄
対応機種/X68000（30可） **¥3,400**

大戦略 III '90

90年代にふさわしくパワーアップされた「大戦略」シリーズ。戦略思考ルーチン、ゲームスピード、コマンド体系、リアルタイムオペレーションなど大幅革新された作品です。

制作／システムソフト
対応機種／ X68000 **¥2,500**

ルーンワース「黒衣の貴公子」

ハイドライドシリーズに続く、新ARPGシリーズ第1弾。綿密に構築された世界「ルーンワース」を舞台に、極めて自由度の高いゲームシステムの中で、興奮の冒険が始まります。

制作／T&Eソフト
対応機種／X68000 **¥700**

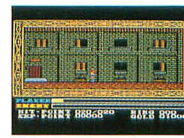
伊忍道　打倒信長

1つのゲームでSLG とRPG、2つのジャンルが楽しめるリコエイションゲームの第3弾。特にRPGの要素が濃い、異色傑作だ!意志を持ったキャラクターが目的に向かって行動を展開。敵を倒して腕を上げ、技を磨いて信長を倒せ!

制作/光栄
対応機種/X68000（30不可） **¥3,400**

ジェノサイド 2

あのズームのゲームがついに名作文庫に登場！特大キャラとハデハデな演出で、68ユーザーのどぎもを抜いた名作アクションゲームだ。MIDIにも対応しているぞ。

制作/ズーム
対応機種/X68000（30不可） **¥2,500**

イース III（ワンダラーズフロムイース）

よりアクション性を増した、これまた、大人気を博したアクション・ロールプレイング。アドルの最後の冒険物語でした。攻撃方法もいっそう多彩になって、時間を感じさせない逸品です。

制作／日本ファルコム
対応機種／X68000（30不可） **¥2,000**

パソコンソフト
自動販売機

TAKERU事務局
〒467 名古屋市瑞穂区苗代町2番1号
ブラザー技術開発センタービル2Ｆ
TEL(052)824-2493（受付時間：月～金 13:00～18:00）

営業所
東京営業所
(03) 5443-4967
大阪営業所
(06) 258-3024

通信販売 1994年4月1日より、送料／手数料が有料になりました。

ソフト名、機種名、メディアのサイズ、住所、氏名、電話番号を明記の上TAKERU事務局まで現金書留でお申し込みください。送料／手数料は、1回のお申し込み総金額が5,000円以上の方は無料。4,900円までの方は500円をいただきます。4,900円までの方は現金500円をプラスしてお申し込みください。誠に勝手ながら、皆様のご理解とご協力の程、お願い申し上げます。

T1002179030689 雑誌 02179-3